SCHOOL OF

CIVIL ENGINEERING

INDIANA

DEPARTMENT OF TRANSPORTATION

JOINT HIGHWAY RESEARCH PROJECT

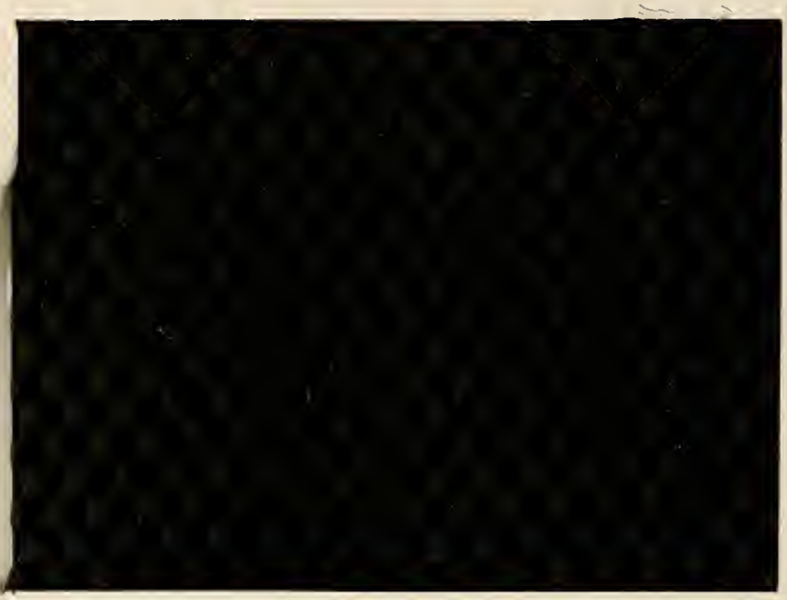
Final Report

FHWA/IN/JHRP-92/21

USER'S MANUAL FOR THE IMPLEMENTATION OF THE INDIANA BRIDGE MANAGEMENT SYSTEM

Lisa C. Gion
Justin Gough
Jeffrey D. Vitale
Kumares C. Sinha
Robert E. Woods

PURDUE UNIVERSITY



JOINT HIGHWAY RESEARCH PROJECT

Final Report

FHWA/IN/JHRP-92/21

USER'S MANUAL FOR THE IMPLEMENTATION OF THE INDIANA BRIDGE MANAGEMENT SYSTEM

Lisa C. Gion
Justin Gough
Jeffrey D. Vitale
Kumares C. Sinha
Robert E. Woods

USER'S MANUAL FOR THE IMPLEMENTATION OF THE
INDIANA BRIDGE MANAGEMENT SYSTEM

Final Report

TO:	Vincent P. Drnevich, Director Joint Highway Research Project	February, 1993 Revised August 12, 1993 Project: C-36-73M
FROM:	Kumares C. Sinha, Research Engineer Joint Highway Research Project	File: 3-4-13

Attached is the Final Report on the HPR study "Implementation of the Indiana Bridge Management System." This reports presents a user's manual for running the project evaluation and program development part of the Indiana Bridge Management System. The research for this report was conducted by Lisa C. Gion, Justin Gough and Jeffrey Vitale, under the direction of myself and Mr. Robert E. Woods of INDOT.

This report is forwarded for review, comments and acceptance by the INDOT and FHWA as fulfillment of the objectives of the project.

Respectfully submitted,

K. C. Sinha

Kumares C. Sinha
Research Engineer

xc:

A. G. Altschaeffl
L. Bandy
M. Byers
P. L. Bourdeau
M. D. Bowman
M. J. Cassidy
L. M. Chang
M. Cohen
E. Cox
S. Diamond
V. P. Drnevich
A. A. Fendrick
J. D. Fricker
K. R. Hoover
R. B. Jacko
L. S. Jones
C. V. Kahl
R. H. Lee
C. W. Lovell
D. W. Lucas
B. Mason
B. G. McCullouch
B. K. Partridge
F. P'Pool
J. A. Ramirez
G. F. Rorbakken
C. F. Scholer
K. C. Sinha
D. L. Tolbert
L. Tucker
R. VanCleave
T. D. White
R. K. Whitford
J. R. Wright

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No. FHWA/IN/JHRP-92/21	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle User's Manual for the Implementation of the Indiana Bridge Management System		5. Report Date November 24, 1992 Revised August 12, 1993 6. Performing Organization Code
7. Author(s) Lisa C. Gion, Justin Gough Kumares C. Sinha and Robert E. Woods		8. Performing Organization Report No. JHRP-92/21
9. Performing Organization Name and Address Joint Highway Research Project Civil Engineering Building Purdue University West Lafayette, IN 47907		10. Work Unit No. 11. Contract or Grant No. Indiana-HPR-2048
12. Sponsoring Agency Name and Address Indiana Department of Transportation State Office Building 100 North Senate Avenue Indianapolis, IN 46204		13. Type of Report and Period Covered Final Report 14. Sponsoring Agency Code

15. Supplementary Notes

Prepared in cooperation with the Federal Highway Administration on HPR past study.

16. Abstract

This report provides a user's manual for the implementation of the project evaluation and program development part of the Indiana Bridge Management System (IBMS). The tools and techniques found in these programs were primarily developed from previous research by the Joint Highway Research Project at Purdue University. The IBMS runs on an IBM-compatible computer under OS/2.

17. Key Words IBMS, decision tree, economic analysis, project ranking, optimization	18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	204	

Form DOT F 1700.7 (8-69)

Final Report

USER'S MANUAL FOR THE IMPLEMENTATION OF THE INDIANA BRIDGE MANAGEMENT SYSTEM

Lisa C. Gion
Justin Gough
Jeffrey D. Vitale
Research Assistants

Kumares C. Sinha
Professor of Civil Engineering

Robert E. Woods
Indiana Department of Transportation

Joint Highway Research Project

Project No: C-36-73M
File No: 3-4-13

Prepared for an Investigation

Conducted

in cooperation with the

Indiana Department of Transportation

and

Federal Highway Administration

Purdue University
West Lafayette, IN 47907

February, 1993
Revised August 12, 1993

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADT	Average Daily Traffic
ADTE	Average Daily Traffic Effectiveness
CEF	Cost Effectiveness Factor
COST	Economic Analysis Program
DC	Deck Condition
DG	Deck Geometry
DTREE	Decision Tree Program
EUAC	Equivalent Uniform Annual Cost
FC	Functional Classification
FHWA	Federal Highway Administration
IBMS	Indiana Bridge Management System
INDOT	Indiana Department of Transportation
IR	Inventory Rating
JHRP	Joint Highway Research Project
MB	Megabyte(s)
NBIS	National Bridge Inventory System
OPT	Optimization Program
RAM	Random-Access Memory
RANK	Project Ranking Program
SUB	Substructure Condition
SUP	Superstructure Condition
VC	Vertical Clearance

CHAPTER 1 INTRODUCTION

1.1 Background

Management of a bridge system involves a complex decision-making process. A comprehensive bridge management system provides a set of tools and techniques that aid in this process by selecting improvement projects, estimating repair costs, prioritizing projects, and so on. Such a system has been developed for the Indiana Department of Transportation (INDOT), based on research performed by the Joint Highway Research Project (JHRP) at Purdue University. For more information regarding the development of these tools and techniques, refer to the reports titled, "The Development of Optimal Strategies for Maintenance, Rehabilitation and Replacement of Highway Bridges," FHWA/IN/JHRP-89/8-13, Volumes 1-6. The titles of all six volumes are listed below:

Volume 1	Elements of Indiana Bridge Management System
Volume 2	A System for Bridge Structural Condition Assessment
Volume 3	Bridge Traffic Safety Evaluation
Volume 4	Cost Analysis
Volume 5	Priority Ranking Method
Volume 6	Performance Analysis and Optimization

1.2 Purpose and Scope of Research

The present study focused on the implementation of the project selection module of the Indiana Bridge Management System (IBMS). The project selection module is a group of decision-making tools originally consisting of three programs: life cycle model, ranking, and optimization. A fourth program has been added to the IBMS, DTREE, which utilizes a decision tree to select projects based on certain traffic, roadway, and bridge properties.

The purpose of this study was to provide a user's manual for running the IBMS package as an integrated tool. It should be noted that the four programs are to be run sequentially. Once the decision tree identifies the possible projects, life cycle costs are computed by the economic analysis program. At this point, a decision can be made only on the basis of cost effectiveness values generated by the COST program. If it is desired to consider additional factors, such as community impact and traffic safety, the next program, RANK, should be used. Similarly, if global optimization is desired, the fourth program, OPT, should be run.

1.3 Report Organization

This report consists of eight chapters. Chapter 2 describes the four programs. Chapter 3 contains basic hardware requirements and instructions regarding the installation, set up, and running of the IBMS. Chapter 4

describes "runfiles," which control option settings and file names. Chapter 5 explains parameter files, which allow the user and the IBMS to communicate with each other. Chapter 6 provides an explanation of "exception files," which allow the user to override information in the bridge database file. Chapter 7 describes the IBMS Tools, which are a set of tools designed to make the IBMS more convenient for the user. Chapter 8 provides an explanation for an example run of the IBMS package. The appendices contain flow charts of program operation, file formats, sample reports, descriptions of the functions and subroutines used in the programs, and other information which may be useful.

CHAPTER 2 PROJECT SELECTION MODULE

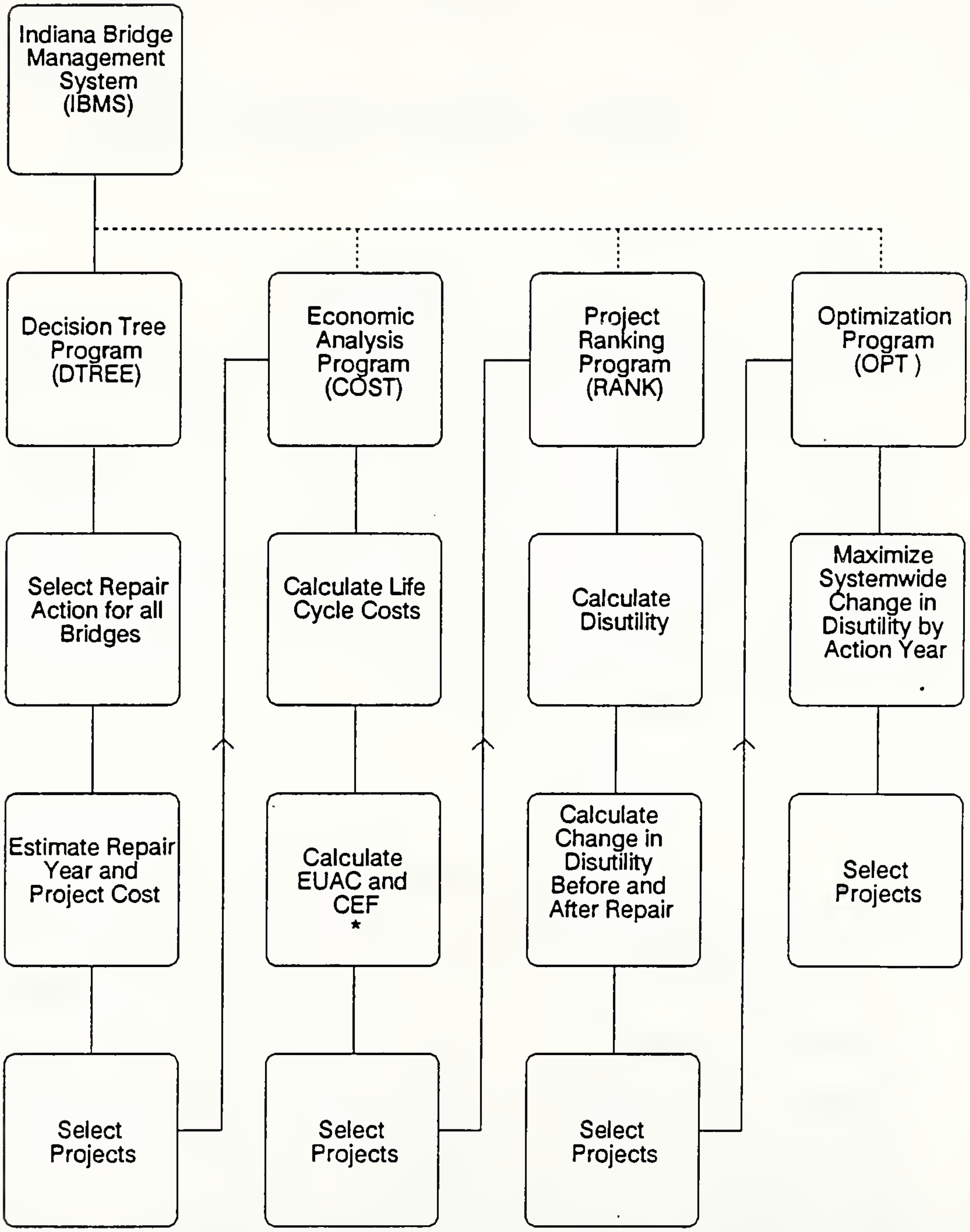
2.1 Introduction

The Indiana Bridge Management System (IBMS) consists of four stand-alone, yet interrelated programs: a decision tree program, an economic analysis program, a project ranking program, and an optimization program. To obtain a full analysis, the programs should be run sequentially; otherwise, unexpected results or errors may occur. A flow chart of the IBMS is illustrated in Figure 2.1.

2.2 Decision Tree Program (DTREE)

The Decision Tree Program (DTREE) is the first program in the IBMS. DTREE analyzes a set of bridges and recommends an action for each bridge. This action consists of three basic choices: do nothing, rehabilitation, and replacement. These three choices can be broken down further into 17 possible repair types; a list can be found in Figure 2.2 and Appendix H.

Decisions are based on bridge condition numbers and bridge, roadway, and traffic characteristics, such as



* EUAC = equivalent uniform annual cost
CEF = cost effectiveness factor

Figure 2.1 IBMS Operation

inventory rating and vertical clearance. Condition numbers rate a bridge's deck, substructure, and superstructure on a scale of 0 to 9, with 9 indicating the maximum rating for a new bridge (Jiang and Sinha 1989). These numbers are updated dynamically using the Markovian process and then rounded to the nearest integer. Once an action is selected, the program also estimates the repair year and project cost.

Repairs are recommended 2, 3, 4 and 5 years in advance from the year of analysis, since this is a typical time period for programming and preliminary engineering. When DTREE recommends a repair, a decision code is saved only when it occurs for the first time. For example, if the analysis year is 1992 and DTREE recommends a "do nothing" in 1994, rehabilitation for 1995 and 1996, and a replacement in 1997, DTREE would only save the decision codes for 1994, 1995, and 1997. If no action is recommended, the action year is automatically set to two years from the analysis year; however, this action will not be processed by subsequent programs.

2.2.1 Program Operation

The flow charts found in Appendices A and D describe basic program operation. A decision tree used to recommend repairs is shown in Figure 2.2.

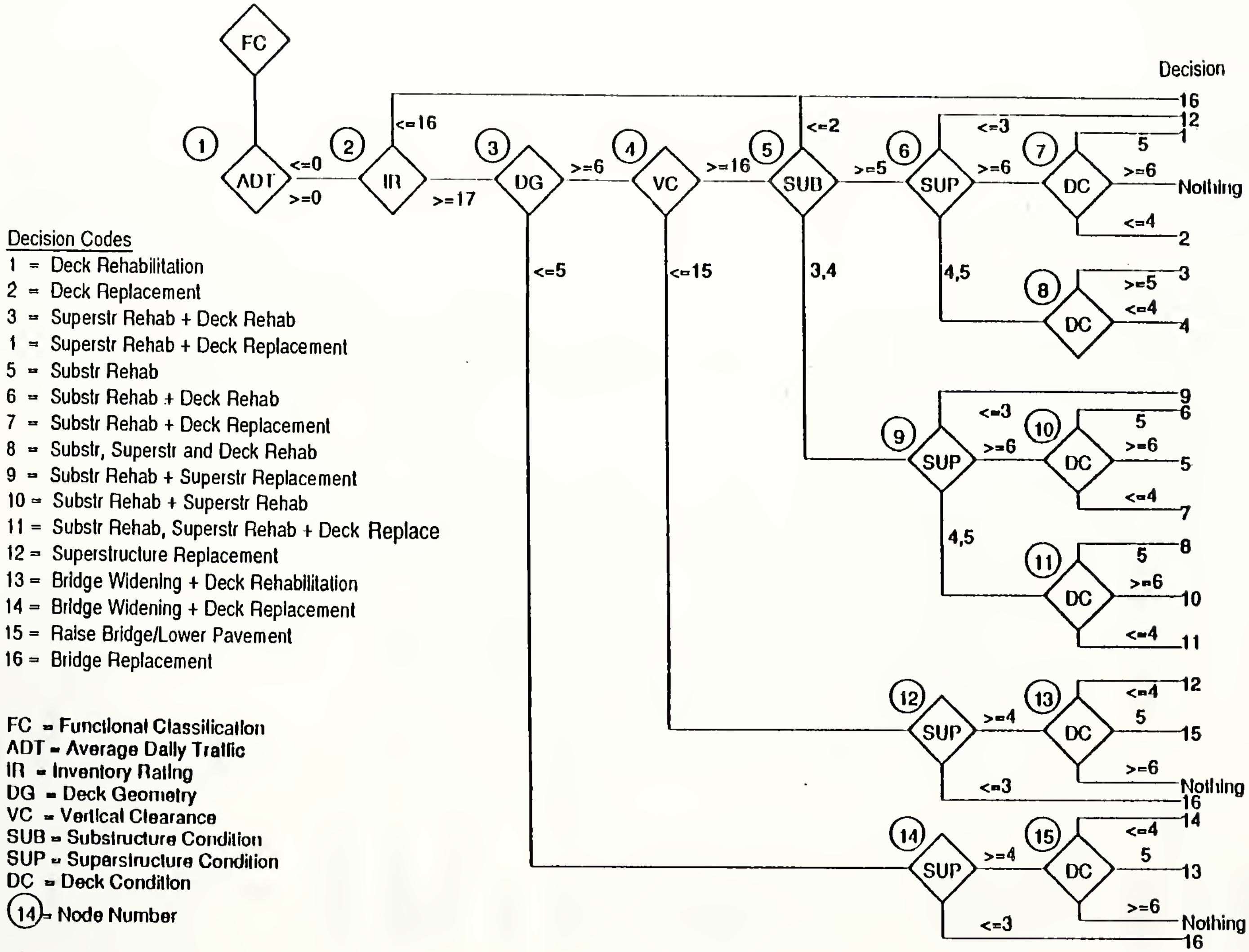


Figure 2.2 Decision Tree for Major Highway Bridges

2.2.2 Input Requirements

2.2.2.1 Data Files

The bridge database file for the IBMS is created by extracting data from the main bridge database at INDOT, which is collected under the National Bridge Inventory System (NBIS). This file must conform to the specified format found in Appendix F.

Exception files, if available, are processed by DTREE, which in turn passes this information to subsequent programs through its output file; refer to Appendix F for more information. Exception files allow the user to modify information in the bridge database, and thus, override decisions made by DTREE and influence how certain actions are performed, such as bridge widening. These files are not essential for normal operation of the system; they merely provide additional flexibility and control. Refer to Chapter 6 for more information regarding exception files. A sample exception file can be found in the directory \bridge\examples.

2.2.2.2 Runfile

The runfile contains all the information regarding option settings and input/output file names used by the IBMS. A default runfile with pre-defined settings is loaded upon program execution; however, the user can specify a runfile that is tailored to his/her own needs. Runfile options

pertaining to DTREE include the following: print text reports, Lotus 123 reports, and log file, check bridge records, report style, and process exception files. These options are explained in depth in Chapter 4. A copy of the runfile can be found in the directories \bridge\run and \bridge\examples.

2.2.2.3 Parameter Files

Parameter files are text files read by the IBMS. These files allow the user to easily modify options, equation constants, weights, conversion factors, etc., rather than searching through the source code. DTREE uses the following parameter files: decision tree, cost, and dollar conversion. Chapter 5 provides an in depth explanation of these files.

2.2.2.4 Keyboard Input

The user will first be prompted for a runfile name. The default runfile can be used or one can be specified. If an external runfile is used, the file extension (e.g., *.run) must be entered. Next, the user will be asked for the analysis year (four digits), the report style, and whether or not exceptions will be used.

2.2.3 Output

If the default runfile is used, the program creates four report files called dtree, with the following extensions: .txt, .prn, .out, and .log. A description of each extension is provided below.

.txt	program report
.prn	program report, set off by commas and quotation marks for a spreadsheet program
.out	bridge information read by subsequent programs
.log	record of program execution

The user can choose from three different report styles: normal, condition, and exception. The normal report contains basic information, such as the bridge number and designation, route number, district and county codes, action year and code, a description of the action, estimated cost, whether or not exceptions were used, and the total number of projects selected for each action. The condition report contains the same information as the normal report; it also provides the deck, substructure, and superstructure condition numbers for the year of inspection, as well as for analysis year. The exception report also contains the same information as the normal report, but also includes exception file data. Note that if exceptions are not used, these columns will be filled with values of zero. Samples of the reports and log file can be found in Appendix G; a sample output file can be found in Appendix F.

2.3 Cost Estimate Program (COST)

Once the initial investment is made to construct a bridge, future funding needs must also be analyzed, since a bridge represents a long-term investment. The Economic Analysis Program (COST) is the second program in the IBMS. This program performs a life cycle cost analysis for each bridge, estimating maintenance, rehabilitation, and replacement costs over the life of a bridge. If no repair is recommended by DTREE (action code 0), COST will not perform a life cycle cost analysis.

To compare the costs of different projects on an equal basis, these costs are converted to an equivalent uniform annual cost (EUAC) in perpetuity. A cost effectiveness factor (CEF) is then calculated, based on ADT, EUAC, and deck area. This factor serves as a normalizing factor, so that bridges with different characteristics and service levels can be compared.

2.3.1 Program Operation

The flow charts found in Appendices B and D describe basic program operation. Derivations for cost equations can be found in the report, FHWA/IN/JHRP-89/11, "The Development of Optimal Strategies for Maintenance, Rehabilitation and Replacement of Highway Bridges, Final Report Vol. 4: Cost Analysis" (Saito and Sinha 1989).

2.3.2 Input Requirements

2.3.2.1 Data Files

An output file from DTREE provides the bridge number, projects selected, exceptions used, and other information; see Appendix F for a complete description. Once the bridge number is known, other information, such as length and width, is found by locating the bridge record in the bridge database file. The same database file and runfile used by DTREE must be used in COST; otherwise, unexpected results or errors may occur. In addition, the same analysis year should be specified.

2.3.2.2 Runfile

Options similar to those mentioned in Section 2.2.2.2 are available in COST. In addition, the user also has the option to calculate maintenance costs and lengthen and widen bridges; refer to Figure 4.1, lines 17 through 19, for more information.

2.3.2.3 Parameter Files

The following parameter files are used by COST: cost, life cycle model, and dollar conversion. Chapter 5 provides an in depth explanation of these files.

<u>2.3.2.4 Keyboard Input</u>

The user will be prompted for a runfile name. The default runfile can be used or one can be specified. If an external runfile is used, the file extension (e.g., *.run) must be entered. Next, the user will be asked for the analysis year (four digits), the report style, and whether or not exceptions will be used.

2.3.3 Output

If the default runfile is used, the program creates four report files called cost, with the following extensions: .txt, .prn, .out, and .log. The user can choose from four different report styles: normal, look ahead, present worth, and present worth look ahead. The normal report contains basic information, such as the bridge number and designation, route number, district and county codes, action year and code, a description of the action, estimated cost, EUAC, CEF, whether or not exceptions were used, and total and average project costs. The look ahead report contains the same information as the normal report, but also estimates future repairs and their costs. The present worth and present worth look ahead reports calculate the present worth for projects. Samples of the reports and log file can be found in Appendix G; a sample output file can be found in Appendix F.

2.4 Project Ranking Program (RANK)

The Project Ranking Program (RANK) is the third program in the IBMS. This program computes the total disutility for a bridge, based on the following criteria: cost effectiveness, bridge condition, safety, and community impact. These four criteria are measured by seven utility functions, which are then weighted and added together to form the total disutility, which ranges from 0 to 100. A score of 0 indicates that the bridge is in near perfect condition, while 100 signifies that the bridge requires immediate repair. Once the disutility is computed for each bridge, the projects can be prioritized. See Figure 2.3 for a breakdown of the ranking process.

2.4.1 Program Operation

The flow charts found in Appendices C and D describe basic program operation. A detailed analysis of the ranking procedure can be found in the report, FHWA/IN/JHRP-89/12, "The Development of Optimal Strategies for Maintenance, Rehabilitation and Replacement of Highway Bridges, Final Report Vol. 5: Priority Ranking Method (Saito and Sinha 1989).

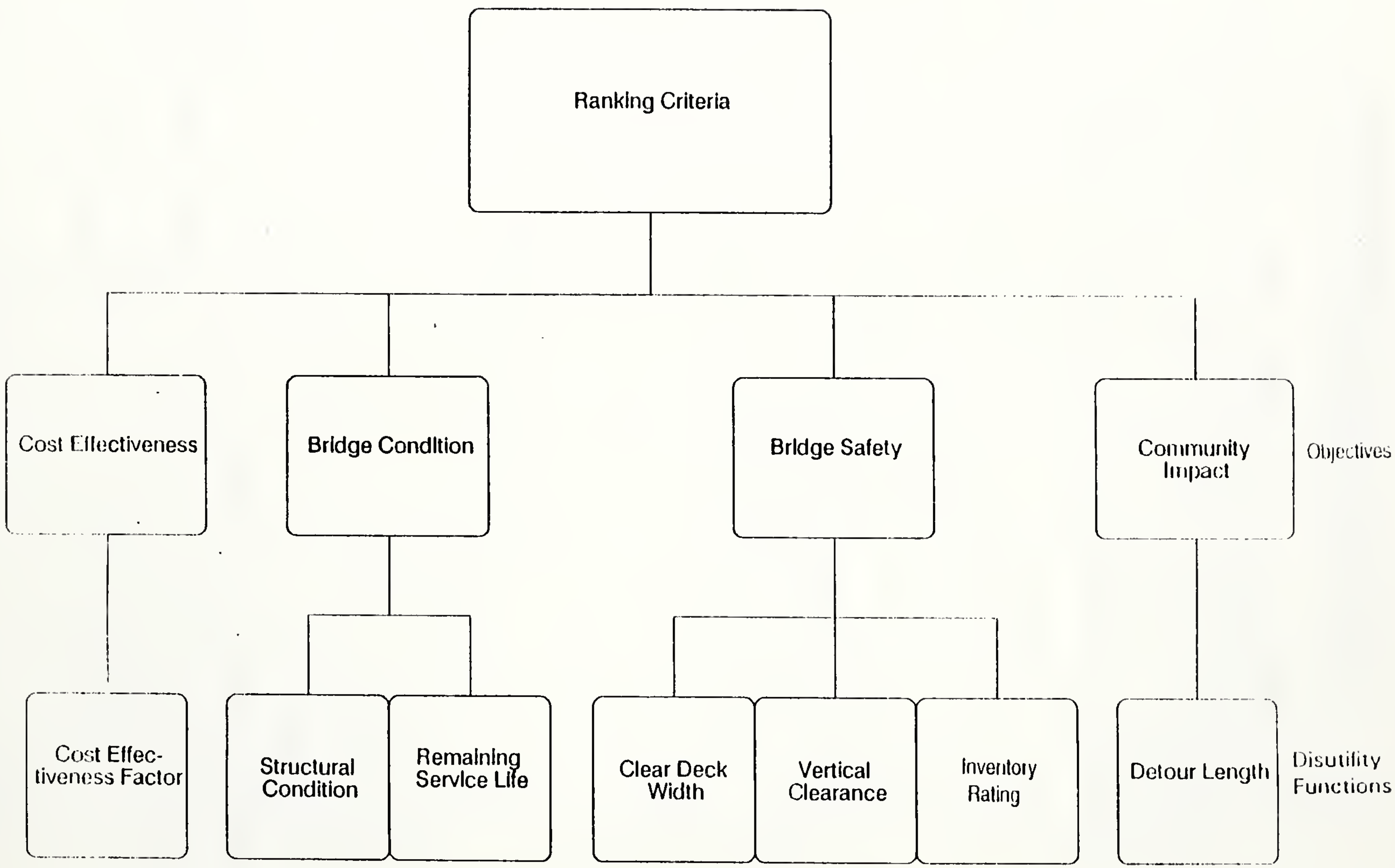


Figure 2.3 Ranking Process

2.4.2 Input Requirements

2.4.2.1 Data Files

An output file from COST provides the bridge number, project costs, and other information; see Appendix F for a complete description. Once the bridge number is known, other information, such as length and width, is found by locating the bridge record in the bridge database file. The same database file and runfile used by DTREE must be used in RANK; otherwise, unexpected results or errors may occur. In addition, the same analysis year should be specified.

2.4.2.2 Runfile

Options similar to those mentioned in Section 2.2.2.2 are available in RANK.

2.4.2.3 Parameter Files

The following parameter files are used by RANK: ranking weight and ranking utility. Chapter 5 provides an in depth explanation of these files.

2.4.2.4 Keyboard Input

The user will be prompted for a runfile name. The

default runfile can be used or one can be specified. If an external runfile is used, the file extension (e.g., *.run) must be entered. Next, the user will be asked for the analysis year (four digits) and the report style.

2.4.3 Output

If the default runfile is used, the program creates four report files called rank, with the following extensions: .txt, .prn, .out, and .log. The user can choose from two different report styles: normal and utility. The normal report contains basic information, such as the bridge number and designation, route number, district and county codes, action year and code, a description of the action, estimated cost, EUAC, whether or not exceptions were used, total disutility before and after a repair is made, the change in disutility, and the number of projects selected. The utility report contains the same information as the normal report, but also includes the disutility values for each disutility function. Samples of the reports and log file can be found in Appendix G; a sample output file can be found in Appendix F.

2.5 Optimization Program (OPT)

The effectiveness of a project is judged by the expected difference in disutility before and after a repair is implemented. The fourth program in the IBMS is the

Optimization Program (OPT), which applies integer linear programming to select projects over a four-year period. The objective function is to maximize the change in disutility subject to available funding. Funding levels are specified by the user.

2.5.1 Program Operation

A description of basic program operation can be found in Appendix E. A detailed analysis of the optimization procedure can be found in the report, FHWA/IN/JHRP-89/13, "The Development of Optimal Strategies for Maintenance, Rehabilitation and Replacement of Highway Bridges, Final Report Vol. 6 (Jiang and Sinha 1989).

2.5.2 Input Requirements

2.5.2.1 Data Files

An output file from RANK provides the bridge number, disutilities, and other information; see Appendix F for a complete description. Once the bridge number is known, other information, such as length and width, is found by locating the bridge record in the bridge database file. The same database file used by DTREE must be used in OPT; otherwise, unexpected results or errors may occur. In addition, the same

analysis year should be specified.

2.5.2.2 Runfile

A runfile is not used by OPT. Instead, input is manually entered by the user.

2.5.2.3 Parameter Files

Parameter files are not used by OPT.

2.5.2.4 Keyboard Input

The user will be prompted for the analysis year (four digits), each optimization year (four digits), and the yearly budget (in thousands of dollars).

2.5.3 Output

The program outputs a year by year listing of the selected projects, one for each year in the analysis period. These listings are in COST output format and are CHOSEN_1.OUT, CHOSEN_2.OUT, CHOSEN_3.OUT, and CHOSEN_4.OUT. Figures 8.22-8.24 show an example of this report.

2.6 Code Mapping

2.6.1 Introduction

All data found in the bridge database file conforms to the standards specified in the Federal Highway Administration's (FHWA) "Recording and Coding Guide for the Structure Inventory and Appraisal of the Nation's Bridges," December, 1988. The IBMS converts these FHWA codes to a simplified set of internal codes to enable faster processing. For example, the functional class of a bridge determines which decision tree DTREE uses to recommend a project. The Recording and Coding Guide includes approximately ten different functional classes; however, it is impractical to have a different decision tree for each functional class. Therefore, the codes are divided into two groups: major and minor highways. A similar process is carried out for other fields in the bridge database. Table 2.1 provides a description of these mapped codes.

When a bridge record is read from the bridge database file, the data is stored in memory. Therefore, when the record is mapped, the codes stored in memory are replaced by the appropriate values; data from the original file remains unchanged.

Some fields in the bridge database record, such as vertical clearance and approach length, are assigned default values, since measurements may be unavailable from field

Table 2.1 Mapped Bridge Fields

Bridge Field	Recording and Coding Guide Item #	Mapped Code[1]	Original Code(s)[1]	Purpose
Functional Class	26	1: Major highway 2: Minor highway	o 1,2,7,11,12,14 o 6,8,9,16,19	select decision tree[2], maintenance cost formula[3]; estimate repair costs[4]; update condition numbers[5]
Vertical Clearance	54a and 54b	Default or Exception Value	0′ 0″, if bridge is over water or actual value in database if over road or railroad	select repair[2]; cost calculations[4]; traffic safety factor[6]
Superstructure Material	43a	1: Steel 2: Concrete	o 3,4,9 o 1,2,5,6,7,8,0	select life cycle model[3]; estimate repair costs[4]; select superstructure type
Superstructure Type	43b	1: RC slab and Box beam 2: Concrete I-beam 3: Steel beam 4: Steel girder	o 1,5,6,20,0 o 4,8,11,12,18, 19,22 o 15,16,17 o 2,3,7,9,10,13, 14,21	calculate superstructure costs[4]
Approach Length	Not available	Default or Exception Value	Not available	calculate approach costs[4]

[1] Refer to bdata.for and map.for for more information.
[2] Refer to Chapter 5, Section 2.
[3] Refer to Chapter 5, Section 4.
[4] Refer to costest.for, Appendix H, and FHWA/IN/JHRP-89/11 (Saito and Sinha 1989) for more information.
[5] Refer to degrade.for and FHWA/IN/JHRP-89/13 (Jiang and Sinha) for more information.
[6] Refer to Chapter 5, Sections 5 and 6.

inspections. These values are assigned at the same time that the FHWA codes are converted to their internal values.

2.6.2 Viewing Mapped Codes

The user can examine the mapped code values by printing the bridge database file with a program (PRBRIDGE.EXE) found in IBMS Tools. This program creates a text file so that the user can view the file with its original codes or with the mapped codes. These two reports can then be compared to see the original and mapped codes.

The user also has the option to check the database file for errors. Fatal errors occur when a bridge record contains bad or missing data, such as condition numbers, needed for program calculations. Any record with one or more fatal errors will be rejected and an error message will be printed. Nonfatal errors occur when data does not conform to the NBIS, but does not affect program operation. Refer to Chapter 7 for more details.

CHAPTER 3 RUNNING THE IBMS

The Indiana Bridge Management System (IBMS) runs on an IBM-compatible computer system. The IBMS package was developed using IBM FORTRAN/2, under the IBM Operating System/2 (OS/2), Standard Edition 1.3. The IBMS Tools, explained in Chapter 7, were written in Microsoft C. Compatibility problems of IBM FORTRAN/2 with OS/2 2.0 require a DOS window for executing DTREE, COST, and RANK. The included source listings of DTREE, COST, and RANK were compiled and executed on FORTRAN/2 with OS/2 1.3. Check for compatibility with individual systems.

3.1 Hardware Requirements

To run the IBMS, the following equipment is necessary: a 386 IBM-compatible computer with a 20 megahertz processor, 4 megabytes (MB) of available memory (RAM), and 80 MB of hard drive space. While the actual IBMS only requires about 3 MB of hard drive space, the commercial software packages (OS/2, Microsoft C, and IBM FORTRAN/2) require hard drive space on the order of 30 to 60 MB.

3.2 Software Requirements

In order to run the IBMS, OS/2 must be present on the computer. If the user plans to modify the source code, IBM FORTRAN/2, and Microsoft C, version 6.0 or greater, must also be installed. To fully utilize the capabilities of the IBMS, the user may wish to install a spreadsheet program, such as Lotus 123, to view reports with the .prn extension.
In addition, a word processor, such as Word Perfect, may be useful so that the reports can be printed.

3.3 Installing the IBMS

Prior to installation of the IBMS, the CONFIG.SYS file must include the following statements so that the programs run properly:

```
LIBPATH=c:\fortran;c:\;
SET PATH=c:\c600\binp;c:\c600\binb;c:\fortran;c:\bridge\prog;c:\;
SET DPATH=c:\c600\binp;c:\c600\binb;c:\fortran;c:\bridge\prog;c:\;
```

Furthermore, a file called STARTUP.CMD, which is placed in the root directory, should contain the following information (IBM 1987):

```
rem FORTRAN Options
rem ============
SET FORTRAN.CER=C:\FORTRAN\FORTRAN.CER
SET FORTRAN.DER=C:\FORTRAN\FORTRAN.DER
SET FORTRAN.ERR=C:\FORTRAN\FORTRAN.ERR
SET FORTRAN.HLP=C:\FORTRAN\FORTRAN.HLP
rem C Options
rem ========
set INCLUDE=c:\C600\INCLUDE
set HELPFILES=c:\C600\HELP\*.hlp
set INIT=c:\c600\INIT
```

```
set TEMP=c:\TMP
SET LIB=C:\FORTRAN\PLIB;C:\C600\LIB
exit
```

Copies of these files can be found in the \bridge\setup directory on the installation disk. Consult the README documentation file in the \bridge\setup directory for complete setup instructions.

To install the IBMS, insert the first installation disk. Type "cd \" at the c: prompt to return to root directory. The prompt should now read c:. To install the programs, type "a:\unzip.exe a:\programs.zip". Repeat the above, replacing programs.zip with fortran.zip, clang.zip, emx_dll.zip each time. The second diskette gets installed similarly, replacing programs.zip with apps_os2.zip, viewer.zip, and config.zip. The third diskette contains runfiles, data files, exception files, and IBMS tools. To properly place them, it is required that the user create the directories \bridge\run, \bridge\bridge, and \bridge\reports. The user can then place these files in the appropriate directories.

3.4 Setting Up the IBMS

Once the files are installed, icons can be created to execute DTREE, COST, RANK, and OPT from the \bridge\prog directory. See Table 3.1 for the location of the four programs. To install icons for the utility programs used for viewing the OPT reports, follow this procedure to create a icon for RANK_1.OUT (for OS/2 2.0):

Table 3.1 IBMS Set-up Procedure

Program title	Path and File Name	Parameters[1]	Working directory	Program type
Decision Tree Program	\bridge\prog\dtree.exe		\bridge\run	default
Economic Analysis Program	\bridge\prog\cost.exe		\bridge\run	default
Project Ranking Program	\bridge\prog\rank.exe		\bridge\run	default
Optimization Program	\bridge\prog\opt.cmd		\bridge\run	default

[1] No information is needed for this field.

3.5 Running the IBMS from the Icon View

1. Double click left mouse button on template folder.

2. Double click left and drag folder with right button to make the IBMS window.

3. Double click right and drag program template into window.

4. In the program template, click once on the right button. Click once on the arrow to the right side of open. Click right once on settings.

5. Type under path and file name:

c:\BRIDGE\VIEWER\BIN\VIEWER.EXE

6. Type (as 1 line) under optional parameters:

--rankfile c:\bridge\reports\rank_1.prn(space)

c:\bridge\viewer\bin4RANK.cfg(space)

c:\bridge\reports\rank_1.prn(space)

c:\bridge\viewer\bin\4RANKlotus.cfg

7. Under working directory:

c:\bridge\viewer\bin

Follow same procedure, successively replacing RANK_1.PRN WITH RANK_2.PRN, RANK_3.PRN, and RANK_4.PRN. For further information on icons and OS/2 windows, consult the IBM OS/2 Standard Edition.

To install icons for CHOSEN_1.OUT, CHOSEN_2.OUT, CHOSEN_3.OUT, and CHOSEN_4.OUT, replace step 6 with

--costfile\bridge\reports\chosen_1.out(space)

c:\bridge\viewer\bin\4COST.cfg(space)

c:\bridge\reports\chosen_1.prn(space)

c:\bridge\viewer\bin\4COSTlotus.cfg

Once the programs are added to the program group, the IBMS is ready to run from the OS/2 Icon View.

3.5.1 DTREE

In the Icon view, move the mouse pointer to DTREE and double-click. A full screen will appear, asking the user for a runfile name. The default runfile can be used or one can be specified. Be sure to type in the file extension (e.g., *.run). Next, enter the analysis year (typically, the current year), the report style, and whether or not exceptions will be used. For more information on customizing a runfile, see Chapter 4. After program execution, an output file (.out) is created, which passes on the necessary information needed by the next program.

3.5.2 COST

Return to the Icon View and double-click the COST program. The user will be prompted for the runfile name; the same runfile specified for DTREE should be used or unexpected results or errors may occur. Next, enter the analysis year, the report style, and whether or not exceptions will be used. After program execution, an output file (.out) is created, which passes on the necessary information needed by the next program.

3.5.3 RANK

Again, return to the Icon View and double-click the RANK program. Enter the runfile name (the same as DTREE and COST), analysis year, and report style. After program execution, an output file (.out) is created, which passes on the necessary information needed by the next program.

3.5.4 OPT

Double-click the OPT program in the Icon View. After the first prompt, enter the analysis year. After each successive prompt, enter the optimization year and its budget, beginning 2 years after the analysis year.

3.6 Running the IBMS from the OS/2 Prompt

To execute the IBMS from the OS/2 prompt, the user must be in the following directory: \bridge\prog. Type "dtree", "cost", "rank", or "optimize". The user will then be prompted for the runfile, analysis year, report style, and whether or not exceptions will be used. When prompted for the runfile, the entire path name must be specified. For example, type c:\bridge\run\runfile.run; otherwise, an error will occur and the program will stop. An example run is provided in Chapter 8, along with an explanation of the results.

3.7 Viewing and Printing Reports

Each program (except OPT) generates a series of files with the following extensions: .txt, .prn, .out, and .log. As OPT is the final program, it generates only a year by year listing of the selected projects and is not required to provide any information in addition to the other reports. See Figure 8.22 through 8.24 for OPT output. A description of each file extension is provided below.

.txt	program report
.prn	program report, set off by commas and quotation marks, for a spreadsheet program
.out	bridge information read by subsequent programs
.log	log of program execution

To view the results, double-click on the IBMS Tools option and then double-click on the View Reports option under the Icon View window. Type in the name of the report, for example, DTREE.TXT or DTREE.LOG. This option calls up a program called LIST.EXE, which is a read-only program that allows the user to view the contents of an ASCII file. This program is configured to read files in the \bridge\reports directory; if the user wishes to view a file in another directory, simply type the full path name, for example, c:\bridge\bridge\bridge.dat.

Another alternative would be to import a .prn report into Lotus 123 or a compatible spreadsheet program. In Lotus 123, import the file as numbers; in Quattro Pro, import as a comma-and-quote delimited file. The other files (.txt, .out, and .log) can also be imported into Lotus 123, but as text, instead of numbers; in Quattro Pro, import as an ASCII text

file. All three file types can be printed within the spreadsheet program. To import OPT output files into the spreadsheet program, see Section 3.4.

In addition to printing reports from a spreadsheet, the user can also use the Print Manager, found in OS/2; for more information, refer to "IBM Operating System/2 Standard Edition Version 1.3 Getting Started," Chapter 6. Reports can also be imported into a word processing program.

CHAPTER 4 RUNFILES AND PROGRAM OPERATION

4.1 Introduction

To provide flexibility, the IBMS offers a variety of options which control program operation. Rather than entering these options each time a program is executed, the system uses a special input file called a "runfile," which centralizes all option settings and file names used by the programs. Hence, minor changes can be made to the runfile without having to reenter the data each time a program is executed. This file can be found in the following directories: \bridge\run and \bridge\examples.

4.2 Default and User Defined Runfile

Each of the main programs in the IBMS is compiled with a set of predefined runfile values, which are initialized at the beginning of program execution. When a program is run, the user is prompted for a runfile name. If "default" or "runfile.run" is entered, the predefined values are used and no external disk file is read. However, if a name other than "default" or "runfile.run" is entered, the program will

attempt to open and read a runfile from disk, which will replace some or all of the default runfile information. If any errors occur when reading this file, the program will stop since it is impossible, or at least unwise, to continue with bad or missing runfile information. The file extension must be specified when a runfile other than the default is used.

To customize a runfile, duplicate the default runfile, found in the directory \bridge\run or \bridge\examples, make the desired changes, and save the file under a different name. For example, the user may wish to use a bridge database file called TEST.DAT in the runfile, rather than BRIDGE.DAT, which is the default name. Thus, modify the runfile by using a text editor, such as the OS/2 System Editor, or a word processor, and type "test.dat" in the appropriate place.

4.3 File Layout

A runfile is divided into two main sections: 1) option settings and 2) input, output, and parameter files. The option settings control program operation, such as report style, and consist of a series of yes/no questions for each program. Reserved fields are provided for future programs. Each option has a one-character field; if more than one character is entered, only the first one is read, while the rest are ignored. Note that all entries for option settings must be in lowercase letters.

The second section of the runfile contains the names of

input, output and parameter files. These names may have up to eight characters, with a three character extension. Each file can be found in a specific directory; therefore the full path name should not be specified. See Table 4.1 for more details. File names are case insensitive; i.e., upper or lowercase can be used. Table 4.2 provides an explanation of the runfile.

Table 4.1 File Specifications

File(s)	Directory	Extension	Default name(s)
Bridge Database	\BRIDGE\BRIDGE	*.dat	bridge.dat
Bridge Index[1]	\BRIDGE\BRIDGE	*.inx	bridge.inx
Runfile	\BRIDGE\RUN	*.run	runfile.run
Exception[2]	\BRIDGE\RUN	*.exp	except.exp except2.exp
Parameter[3]	\BRIDGE\PARAM	*.par	dtree.par cost.par life.par dollar.par weight.par utility.par
Output	\BRIDGE\REPORTS	*.out[4]	dtree.out cost.out rank.out chosen_1.out ... chosen_4.out
Reports	\BRIDGE\REPORTS	*.txt[5,6] *.prn[5,7]	dtree.txt dtree.prn cost.txt cost.prn rank.txt rank.prn
Log	\BRIDGE\REPORTS	*.log[5,8]	dtree.log cost.log rank.log

[1] This field is ignored in the current IBMS; future versions may use this field to allow random access of bridge records.
[2] See Chapter 6 for more information.
[3] See Chapter 5.
[4] *.out files provide information read by the programs. See Appendix F for examples.
[5] See Appendix G for examples.
[6] *.txt reports can be examined via the IBMS Tools. See Chapter 7.
[7] *.prn files can be imported into any Lotus 123 compatible spreadsheet program.
[8] *.log files provide a record of program execution.

```
#run_file
#
#
#
#
y               ] All: Print text reports? (y/n)
y               ] All: Print 123 reports? (y/n)
y               ] All: Print log file? (y/n)
y               ] All: Check bridge records? (y/n)
n               ] All: Process exceptions? (y/n)
c               ] Dtree: Report Style (n)ormal (c)ondition (e)xception
n               ] Dtree: Use second exception file? (y/n)
                ] Dtree:
                ] Dtree:
                ] Dtree:
l               ] Cost: Style (n)orm pres(w)orth (l)ookahead (p)wlookahead
y               ] Cost: Do maintenance cost calculations? (y/n)
n               ] Cost: Lengthen Bridges with exceptions and replace? (y/n)
n               ] Cost: Widen Bridges with exceptions and rehab? (y/n)
                ] Cost:
u               ] Rank: Report style (n)ormal (u)tility
                ] Rank:
                ] Rank:
                ] Rank:
                ] Rank:
                ] Opt:
                ] Opt:
                ] Opt:
                ] Opt:
                ] Opt:
                ] reserved
                ] reserved
                ] reserved
                ] reserved
                ] reserved
c
                Bridge database files
bridge.dat      ] Bridge data file from \bridge\bridge
default         ] Bridge index file from \bridge\bridge
except.exp      ] Primary exception file
except2.exp     ] Secondary exception file
c
                Decision tree program files
dtree           ] File prefix for .out .txt .prn .log output files
dtree.par       ] Parameter file: defines decision criteria (must have .par)
c
                Economic analysis program files
dtree.out       ] Input file for cost analysis
cost            ] File prefix for .out .txt .prn .log output files
life.par        ] Life cycle model parameters (must have .par)
cost.par        ] Cost estimate coefficient parameter file
c
                Ranking program files
cost.out        ] Input file for ranking
rank            ] File prefix for .out .txt .prn .log output files
weight.par      ] Ranking weights (must have .par)
utility.par     ] Disutility function parameters (must have .par)
c
                Optimization program files
rank.out        ] Input file for optimization
opt             ] File prefix for .txt .prn .log output files
opt.par         ] Optimization parameters (must have .par)
c
                Other files
dollar.par      ] Interest rate and $ conversion factors (must have .par)
default         ] Log File (overrides log file name from above)
```

Note: Line numbers are provided for reference only; they do not appear in the actual file.

Figure 4.1 Runfile Listing

Table 4.2 Description of Runfile

Line Number	Entry	Description[1]
1	#run_file	o File type identifier. o This line cannot be changed; otherwise, the program will not run.
6	y or n	o If "y", text reports are printed for each program. o Applicable to all programs.
7	y or n	o If "y", a file is created, which can be read by Lotus 123 or a compatible spreadsheet program. o Use the import command to load the file. o Applicable to all programs.
8	y or n	o If "y", a record of each program run is printed. o Useful for pinpointing any errors that may occur during program execution. o Applicable to all programs.
9	y or n	o If "y", records read from the bridge database are checked for errors. o If bad bridge records are processed (set to "n"), program operation will be unpredictable. o All errors, fatal and non-fatal, are flagged and listed in DTREE's logfile.[2] o Prbridge.exe may also be used to check bridge database files.[3] o Applicable to all programs.
10	y or n	o If "y", system will attempt to find exception records for each bridge. o If "n", system will not process any exception records, including those from the secondary exception file.[4]

Table 4.2, continued

Line Number	Entry	Description[1]
11	n, c, or e	o Report styles: (n)ormal, (e)xception, and (c)ondition numbers.[5] o Applicable only to DTREE.
12	y or n	o If "y", system will attempt to open and use a secondary exception file.[4] o Applicable only to DTREE.
13 - 15	-	o Reserved for future use. o Applicable only to DTREE.
16	n, l, p, or w	o Report styles: (n)ormal; (l)ook ahead; normal, present (w)orth; and look ahead, (p)resent worth.[5,6] o Applicable only to COST.
17	y or n	o If "y", maintenance costs are computed in the life cycle analysis. o To disable, set option to "n". o Applicable only to COST.
18	y or n	o If "y", bridges will be lengthened when an exception is specified or a replacement is recommended. o If "n", all bridge lengths remain unchanged.[4] o Applicable only to COST.
19	y or n	o If "y", bridges will be widened when an exception is specified or a widening option is recommended. o If "n", all bridge widths remain unchanged.[4] o Applicable only to COST.
20	-	o Reserved for future use. o Applicable only to COST.
21	n or u	o Report styles: (n)ormal and dis(u)tility values.[5] o Applicable only to RANK.

Table 4.2, continued

Line Number	Entry	Description[1]
22 - 25	-	o Reserved for future use. o Applicable only to RANK.
26 - 30	-	o Reserved for future use. o Applicable only to OPT.
31 - 35	-	o Reserved for future use.
38 - 41	files	o Bridge database files.
44 - 45	files	o Decision tree program files.
48 - 51	files	o Economic analysis program files.
54 - 57	files	o Ranking program files.
60 - 62	files	o Optimization program files.
65 - 66	files	o Miscellaneous files.

Note: all option settings must be typed in lowercase letters; file names may be typed in upper or lowercase.

When making modifications, avoid using tabs and do not add or remove text lines; otherwise, program results may be unreliable.

[1] Refer to Table 4.1 for more information.
[2] Fatal errors occur when a bridge record contains bad or missing data, such as condition numbers, needed for program calculations. Any record with one or more fatal errors will be rejected and an error message will be printed. Nonfatal errors occur when data does not conform to the NBIS, but does not affect program operation.
[3] See Chapter 7, Section 2.1.
[4] See Chapter 6.
[5] See Appendix G for examples.
[6] The first two reports styles convert costs from the base year to the year of analysis; the last two calculate present worth costs. See Chapter 5, Section 7 and econ.for for more information.

CHAPTER 5 PARAMETER FILES

Parameter files are the primary link between the IBMS and the user. These files allow the user to easily modify equation constants, coefficients, conversion factors, and other information, without having to search through the source code.

All parameter files are text files that contain both data and comments, and are read with standard FORTRAN or C statements found in the source code. These files may be edited with any text editor, such as the OS/2 System Editor, or word processing program. Unlike the source code, parameter files do not need to be re-compiled when modifications are made.

5.1 Parameter File Format

Certain features are common to all parameter files in the IBMS: parameter file type, comment area, data lines, and comment lines.

5.1.1 Parameter File Type

The first line of the file is a continuous string of characters, no more than 80 characters long, which identifies the file type. Programs using parameter files will check this string to ensure that the parameter file is of the proper type; if not, an error will be flagged. An example of a file type is written as follows: #ranking_weight_parameter_file. These file type identifiers must be identical to those expected by the IBMS, or programs that use parameter files will be unable to load them. Therefore, the user should avoid changing the first line of the file when editing the parameter file.

5.1.2 Comment Area

The next 4 lines of the parameter file contain information, such as instructions and comments, that may be helpful to the user. Each line must have at least one character or space in the first position, and no more than 80 characters per line. For example, a typical comment area may include the following information:

```
# This file contains weights for use in the ranking process.
# Each weight must be a non-negative number.
# Weights will be normalized to sum to 1 when the program is run.
# Be careful not to delete or add lines when editing this file.
```

The remainder of the parameter file consists of data and/or comment lines. No lines may be added or deleted from the file, or else programs which use these parameter files will either not run or produce unreliable results.

5.1.3 Data Lines

A data line includes two fields: data and comment. The data field contains information processed by the programs and is found in positions 1 through 20. The comment field consists of the remaining 60 characters.

To avoid program termination, certain restrictions apply. Programs reading information from the data field may look at all 20 characters; therefore, it is important that the data and comment fields do not overlap. Overlap can be avoided by ensuring that all 20 positions in the data field are filled with either data or blank spaces. Avoid using tabs to fill in the data field; while the file may look acceptable on the screen, a Fortran program will interpret each tab as only one character position. Therefore, if tabs are used in the data field, the program may read information from the comment field, causing an error to occur and the program to stop.

5.1.4 Comment Lines

These lines act as spacers in the parameter file, making it more readable and understandable. As mentioned previously, no lines may be added or removed. Comment lines must include at least one character or space, and no more than 80 characters. Note that some listings may appear to have blank lines; in reality, spaces have been inserted. To avoid

program errors, add at least one character. Examples include the following:

```
1   5    10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+
                   ] Enter Bridge database file below
c
c
       ---------- Section 2 of Parameter File ----------
c
```

5.2 Decision Tree Parameter File

This parameter file defines the criteria for each branch of the decision tree. The decision tree consists of a series of nodes, where a decision is made at each one; based on that decision, the tree proceeds to another node. After a series of decisions are made, the tree recommends a project or action for the bridge. Note that this parameter file is used only by DTREE.

A decision tree for major highway bridges was illustrated in Figure 2.2. Each diamond on the diagram symbolizes a node of the decision tree; up to 16 nodes can be defined. Each node represents a property of the bridge that is used to make a decision; for example, node 3 (DG) stands for the deck geometry code, Item 68 in the Recording and Coding Guide. In addition, each node has two numbers associated with it: a "high value" and a "low value." The former value is represented by a greater than or equal to sign (>=) and the latter by a less than or equal to sign (<=).

Note that the structure of the tree is fixed; nodes that

have two possible branches may never be converted to nodes with three branches. However, by setting the high and low values properly, a node with three branches may be manipulated to behave as if it had only two branches. To avoid errors in the decision process, a node with only two branches must have high and low values that are consecutive integers; nodes with three branches do not require consecutive integers. Examples for a major highway are given in the following section.

5.2.1 Case 1: Node 3 - DG (Deck Geometry)

The high and low values for this node are set at 6 and 5, respectively. If this node is encountered in the decision making process for a bridge, then its deck geometry code is inspected in the bridge database record. If the deck geometry code is:

>= 6, the branch proceeds to node 4, VC
<= 5, the branch proceeds to node 14, SUP

If the high and low values were specified as 6 and 4, respectively, an error would occur if the bridge's deck geometry code was 5, since 5 is neither less than or equal to 4 or greater than or equal to 6. Consequently, a node with only two branches must have consecutive integers for its high and low values to avoid errors in the decision process.

5.2.2 Case 2: Node 9 - SUP (Superstructure Condition)

The high and low values for this node are specified as 6 and 3, respectively. If the decision making process for a bridge reaches this node, its superstructure condition number is inspected. If this number is:

>= 6, the tree proceeds to node 10, DC
<= 3, an action of 9 is recommended
= 4 or 5, the tree branches to node 11, DC

Since these values are not consecutive integers, a node with three branches exists. It is possible to limit the node to only two branches; however, the elimination of a branch may produce unpredictable results. For example, if the high and low values are set to 4 and 5, respectively, it is impossible for the decision tree to reach node 11.

5.2.3 Defining a Decision Tree

As illustrated in the examples above, each node is assigned two numbers that dictate the decision process at that point. A set of 16 pairs of high and low values describes a decision tree; up to four sets of decision trees may be defined in the IBMS. Each set applies to all bridges with a given mapped functional class code. This code is a number from 1 to 4, which is derived from the functional class code of a bridge, as defined in Item 5 of the Recording and Coding Guide. When a bridge record is read, its functional class code is converted from the Recording and Coding Guide standard

to one of these four mapped codes. At present, only two mapped functional class codes are defined, 1 and 2, which correspond to major and minor highway bridges, respectively. For more information, see Chapter 2, Section 6 and Table 2.1.

Each set of decision tree criteria is defined in the parameter file as a block of 16 pairs of high and low values. Additional blocks may be added to the end of the file; however, all 16 pairs of numbers and all comment lines must be included for proper program execution. To add additional blocks, the programs MAP.FOR and BDATA.FOR must be modified and all system programs must be re-compiled to incorporate this additional information. For information on compiling the programs, refer to Appendix I.

Decision tree criteria sets are presently defined for major and minor highway bridges. High and low values for these sets may be changed by editing the default decision tree parameter file and specifying a new name, such as dtree2.par, in the runfile. When making modifications to these two sets, the program does not have to be re-compiled.

5.2.4 Default Decision Tree

A diagram of the default decision tree for mapped functional class 1 (major highway) was provided in Chapter 2; see Figure 2.2. A program listing of the file is provided in Figure 5.1; Table 5.1 provides a general overview of the decision tree parameter file. Note that the values defining

```
#decision_tree_parameter_file
#
#
#
#
x                  ] reserved
x                  ] reserved
x                  ] reserved
x                  ] reserved
c
1                  Major Highway Functional Class (Item 26: 1, 2, 7, 11, 12, or 14)
c
0                  ] Low  for node 1 ADT
0                  ] High for node 1
c
16                 ] Low  for node 2 Inventory Tons
17                 ] High for node 2
c
5                  ] Low  for node 3 Deck Geometry
6                  ] High for node 3
c
15                 ] Low  for node 4 Vertical Clearance (ft)
16                 ] High for node 4
c
2                  ] Low  for node 5 Substructure Condition
5                  ] High for node 5
c
3                  ] Low  for node 6 Superstructure Condition (a)
6                  ] High for node 6
c
4                  ] Low  for node 7 Deck Condition (a)
6                  ] High for node 7
c
4                  ] Low  for node 8 Deck Condition (b)
5                  ] High for node 8
c
3                  ] Low  for node 9 Superstructure Condition (b)
6                  ] High for node 9
c
4                  ] Low  for node 10 Deck Condition (c)
6                  ] High for node 10
c
4                  ] Low  for node 11 Deck Condition (d)
6                  ] High for node 11
c
3                  ] Low  for node 12 Superstructure Condition (c)
4                  ] High for node 12
c
4                  ] Low  for node 13 Deck Condition (e)
6                  ] High for node 13
c
3                  ] Low  for node 14 Superstructure Condition (d)
4                  ] High for node 14
c
4                  ] Low  for node 15 Deck Condition (e)
6                  ] High for node 15
c
0                  ] Low  for node 16 Not Used
0                  ] High for node 16
c
2                  Minor Highway Functional Class (Item 26: 6, 8, 9, 16 or 19)
c
0                  ] Low  for node 1 ADT
0                  ] High for node 1
c
16                 ] Low  for node 2 Inventory Tons
```

Figure 5.1 Decision Tree Parameter File Listing

```
17                    ] High for node 2
c
5                     ] Low  for node 3 Deck Geometry
6                     ] High for node 3
c
15                    ] Low  for node 4 Vertical Clearance (ft)
16                    ] High for node 4
c
2                     ] Low  for node 5 Substructure Condition
5                     ] High for node 5
c
3                     ] Low  for node 6 Superstructure Condition (a)
6                     ] High for node 6
c
4                     ] Low  for node 7 Deck Condition (a)
6                     ] High for node 7
c
4                     ] Low  for node 8 Deck Condition (b)
5                     ] High for node 8
c
3                     ] Low  for node 9 Superstructure Condition (b)
6                     ] High for node 9
c
4                     ] Low  for node 10 Deck Condition (c)
6                     ] High for node 10
c
4                     ] Low  for node 11 Deck Condition (d)
6                     ] High for node 11
c
3                     ] Low  for node 12 Superstructure Condition (c)
4                     ] High for node 12
c
4                     ] Low  for node 13 Deck Condition (e)
6                     ] High for node 13
c
3                     ] Low  for node 14 Superstructure Condition (d)
4                     ] High for node 14
c
4                     ] Low  for node 15 Deck Condition (e)
6                     ] High for node 15
c
0                     ] Low  for node 16 Not Used
0                     ] High for node 16
```

Note: Line numbers are provided for reference only; they do not appear in the actual parameter file.

When making modifications, avoid using tabs and do not add or remove text lines; otherwise, program results may be unreliable.

Figure 5.1, continued

Table 5.1 Decision Tree Parameter File Field Descriptions

Line Number	Entry	Description
1	#decision_tree_parameter_file	o Parameter file type identifier. o This line cannot be changed; otherwise, the program will not run.
2 - 10	Comments	o Do not remove any comment lines from the file; these lines may contain instructions or information for the user.
11	1	o Mapped functional class code for the decision criteria set. o Used internally to locate and choose the correct decision tree for a bridge. o Do not change this field.
13 - 14	Number Number	o High and low values for node 1 of the decision tree.
16 - 17	Number Number	o High and low values for node 2.
58 - 59	0 0	o High and low values for node 16. o Note that this node does not exist in the default decision tree (values are therefore set to 0); these values are included for future expansion.
61	2	o Mapped functional class code for the second set of decision criteria. o Used internally to locate and choose the correct decision tree for a bridge. o Do not change this field.
62 +	Number Number	o These lines describe the second set of the decision tree criteria.

When making modifications, avoid using tabs and do not add or remove text lines; otherwise, program results may be unreliable.

the decision tree criteria are the same for both major and minor highways; the user can modify these values as needed.

If modifications are made to the parameter file, extra comment lines may not be added to the end of the file because the subroutine (GETDECIS.FOR) that reads the decision tree information may mistake them for a third set of numbers and cause unpredictable results. If errors occur when reading this file, check the format of the parameter file; the listing provided in Figure 5.1 can be used for comparison.

5.3 Cost Parameter File

The cost estimate parameter file contains the information needed to compute replacement and rehabilitation costs for bridge projects. While the functional forms of the equations used to compute these costs are fixed, the constants, coefficients, and powers of the equations may be modified within the parameter file. A program listing of the file is shown in Figure 5.2; Table 5.2 provides an overview of the parameter file. Note that this parameter file is used only in the COST program. Derivations for cost equations can be found in the report, FHWA/IN/JHRP-89/11, "The Development of Optimal Strategies for Maintenance, Rehabilitation and Replacement of Highway Bridges, Final Report Vol. 4: Cost Analysis" (Saito and Sinha 1989).

```
#cost_parameter_file
#
#
#
#
c                    Rehabilitation (volume 4 section 2.3)
c                    --------------
c
c                    Option 1) Deck Replace
c
c                    cost = width * length * a1 * (1+a2)/1000
31.185               ] a1
0.8103               ] a2
c
c                    Option 2) Deck Reconstruction and Overlay
c
7.0                  ] b1 % of surface patched for a major highway
12.0                 ] b2 % of surface patched for a minor highway
c
c                    Patching Costs (da = deck area, pa = patch %)
15.0                 ] c1 patch % cut-off
500.0                ] c2 deck area break 1
2000.0               ] c3 deck area break 2
13.74                ] d1 unit cost 1 ($/ft2) if pa < c1 and da < c2
9.09                 ] d2 unit cost 1 ($/ft2) if pa < c1 and c2 <= da <c3
5.73                 ] d3 unit cost 1 ($/ft2) if pa < c1 and da >= c3
16.09                ] d4 unit cost 1 ($/ft2) if pa >= c1 and da < c2
10.11                ] d5 unit cost 1 ($/ft2) if pa >= c1 and c2 <= da <c3
8.11                 ] d6 unit cost 1 ($/ft2) if pa >= c1 and da >= c3
c
c                    Deck Reconstruction Cost rb=unitcost1*area/1000
100.0                ] e1 recon cost break point
1.2331               ] f1 unit cost 2 if (rb < e1)
0.9311               ] f2 unit cost 2 if (rb >= e1)
c                    Total rehab cost = rb*(1 + unitcost2)
c
c
c                    Bridge Replacement (volume 4 section 2.2)
c                    ------------------
c
c                    Superstructure cost
c
c                    RC Slab or Box Beam cost = g1*len**g2*wid**g3
0.0137               ] g1
1.001                ] g2
1.161                ] g3
c                    Concrete I-Beam cost = g4*len**g5*wid**g6
0.033                ] g4
0.907                ] g5
1.043                ] g6
c                    Steel Beam cost = g7*len**g8*wid**g9
0.0102               ] g7
1.12                 ] g8
1.117                ] g9
c                    Steel Girder cost = g10*len**g11*wid**g12
0.0885               ] g10
0.906                ] g11
0.747                ] g12
c
c                    Substructure Cost
c
c                    Solid Stem cost = h1*len**h2 * wid**h3 * vclear**h4
0.00506              ] h1
0.744                ] h2
1.205                ] h3
0.515                ] h4
```

Figure 5.2 Cost Estimate Parameter File Listing

```
c                     Pile Type cost = h5*len**h6 * wid**h7 * vclear**h8
0.00354               ] h5
0.744                 ] h6
1.205                 ] h7
0.515                 ] h8
c
c                     Approach Costs = i1 * approachlength**i2
0.769                 ] i1
0.823                 ] i2
c
c                     Other Costs[1] = j1*len**j2 * wid**j3
0.0721                ] j1
0.696                 ] j2
0.932                 ] j3
c
c                     Total Replace Cost = supcost+subcost+appcost+othercosts
c
c
c                     Miscellaneous default costs
c                     ---------------------------
c
25.0                  ] Default cost of substructure rehabilitation ($1000's)
350.0                 ] Default cost of raising/lowering pavement ($1000's)
200.0                 ] Default cost of culvert replacement ($1000's)
```

Note: Line numbers are provided for reference only; they do not appear in the actual parameter file.

When making modifications, avoid using tabs and do not add or remove text lines; otherwise, program results may be unreliable.

[1] "Other" costs include structure, mobilization/demobilization, traffic control, demolition, and miscellaneous costs, expressed in terms of deck area (Saito and Sinha 1989)

Figure 5.2, continued

Table 5.2 Cost Parameter File Field Descriptions

Line Number	Entry	Description
1	#cost_parameter_file	o File type identifier. o This line cannot be changed; otherwise, the program will not run.
2 - 11	Comments	o Do not remove any comment lines from the file; these lines may contain instructions or information for the user.
12 - 13	a1 a2	o Coefficients for deck replacement calculations. o Real numbers of up to 20 digits may be specified, but the actual precision is limited to the number of digits in the mantissa of a real number, which is dictated by the Fortran compiler used.
17 - 18	b1 b2	o Specifies the estimated patching percentages for major and minor highway bridges.
21 - 29	c1...c3 d1...d6	o Constants for calculation of "unit cost 1." o See costest.for for more information.
32 - 34	e1 f1 f2	o Constants for calculating "unit cost 2."
44 - 46	g1...g3	o Coefficients for calculation of bridge replacement costs for RC slab or box beam bridges.
48 - 50	g4...g6	o Bridge replacement cost coefficients for I-beam bridges.
52 - 54	g7...g9	o Cost coefficients for steel beam bridge replacements.
56 - 58	g10...g12	o Coefficients for steel girder replacements.
63 - 66	h1...h4	o Substructure cost coefficients for solid stem bridges.
68 - 71	h5...h8	o Substructure cost coefficients for pile type bridges.
78 - 80	i1 i2	o Parameters used to estimate miscellaneous (other) costs.
88	25.0	o Default cost (in thousands) for substructure rehabilitation.
89	350.0	o Default cost (in thousands) for raising/lowering pavement.
90	200.0	o Default cost (in thousands) for culvert replacement. o Not used in current IBMS version.

When making modifications, avoid using tabs and do not add or remove text lines; otherwise, program results may be unreliable.

5.3.1 Rehabilitation

There are two main types of rehabilitation: 1) deck replacement and 2) deck reconstruction and overlay. Costs for the first type (action codes 2, 4, 7, and 11) are a function of the deck area, while costs for the second type (action codes 1, 3, 6, 8,and 10) are determined by the patching percentage of the bridge deck area.

Other costs, such as superstructure or substructure repairs, may be derived from these two main rehabilitation types. For example, an action code of 3 specifies a superstructure rehabilitation and deck rehabilitation. To find the cost of this project, the cost of deck rehabilitation is first calculated from the equation specified in the parameter file; next, an additional 5% is added to include the superstructure cost. All other subtype costs are computed in a similar fashion; refer to Appendix H for further information.

5.3.2 Bridge Replacement

The cost of a bridge replacement project (action codes 9, 12, 13, 14, and 16) is a function of the bridge's deck area, superstructure and substructure types, and approach length.

5.3.3 Miscellaneous Default Costs

Some repair codes (5, 15, and 17) depend on neither replacement nor rehabilitation costs; these costs are assigned default values in the programs. Note that at the present time, DTREE will never produce a recommendation for repair code 17; however, a default cost is provided for future expansion. Refer to Appendix H for more details.

5.4 Life Cycle Model Parameter File

To determine equivalent uniform annual cost (EUAC) calculations for bridge projects, a life cycle model must be defined, which allows the system to predict future repair activities. Different models exist for each type of bridge, such as steel or concrete. These models will specify the necessary repairs to perform, as well as when these repairs probably will occur. In addition, a method for estimating maintenance costs for the time interval between repairs must be defined for each bridge type.

The life cycle parameter file allows the user to define life cycle models for different superstructure types. The current IBMS defines only two types: steel (code 1) and concrete (code 2), which are the first and second models in the parameter file, respectively. When superstructure types are added, the third life cycle model should correspond to superstructure type 3, the fourth model to type 4, and so on. Note that when adding a type to the parameter file, the programs BDATA.FOR and MAP.FOR must be modified and the COST

program re-compiled.

A listing of the file is included in Figure 5.3; Table 5.3 provides an overview of the life cycle parameter file. Note that this parameter file is used only in the COST program.

5.4.1 Developing the Life Cycle Models

Certain information is required to develop the models: 1) bridge types, 2) life span, 3) rehabilitation types and times, and 4) maintenance costs.

5.4.1.1 Bridge Types

The bridge superstructure type determines which cost model is used. As mentioned above, only two superstructure types are defined for the IBMS: steel and concrete. Repair types and times also depend on the superstructure type. While repair costs also depend on other bridge properties, the superstructure type alone determines when these repairs are done.

5.4.1.2 Life Span

The average expected life span for each bridge type must be determined. See Table 5.4 for INDOT's estimated bridge design life.

```
#life_cycle_model_parameter_file
#Describes life cycle model for each bridge type.
#
#
#
.                       ] reserved
.                       ] reserved
.                       ] reserved
.                       ] reserved
.                       Steel Superstructure (item 43a code= 3,4, or 9)
1                       ] bridge type code
65                      ] life span of bridge (integer years)
20                      ] time after construction for rehab1 (years)
35                      ] time for rehab2 (set to 0 if no second rehab)
50                      ] time for rehab3 (0 if no third rehab)
0                       ] time for rehab4 (0 if not needed)
3                       ] type code for rehab 1
2                       ] type code for rehab 2 (ignore if no second rehab)
3                       ] type code for rehab 3 (ignore if no third rehab)
0                       ] type code for rehab 4 (ignore if no fourth rehab)
.                       Maint fields are ignored if maint cost calcs are disabled
0.8504                  ] slope for major hwy maint estimates
945.72                  ] constant for major hwy maint estimates
3.0                     ] % increase of replace costs at end of life span for major hwys
5                       ] yrs after repair when % increase begins for major hwys
1.241                   ] slope for minor hwy maint estimates
878.59                  ] constant for minor hwy maint estimates
3.0                     ] % increase of replace costs at end of life span for minor hwys
5                       ] yrs after repair when % increase begins for minor hwys
r                       ] reserved for future use
r                       ] reserved for future use
r                       ] reserved for future use
r                       ] reserved for future use
r                       ] reserved for future use
.
.                       Concrete Superstructure (item 43a code = 1,2,5,6,7,8, or 0)
2                       ] bridge type code
50                      ] life span of bridge (integer years)
20                      ] time after construction for rehab1 (years)
35                      ] time for rehab2 (set to 0 if no second rehab)
0                       ] time for rehab3 (0 if no third rehab)
0                       ] time for rehab4 (0 if not needed)
3                       ] type code for rehab 1
3                       ] type code for rehab 2 (ignore if no second rehab)
0                       ] type code for rehab 3 (ignore if no third rehab)
0                       ] type code for rehab 4 (ignore if no fourth rehab)
.                       Maint fields are ignored if maint cost calcs are disabled
0.8504                  ] slope for major hwy maint estimates
945.72                  ] constant for major hwy maint estimates
3.0                     ] % increase of replace costs at end of life span for major hwys
5                       ] yrs after repair that % increase begins for major hwys
1.241                   ] slope for minor hwy maint estimates
878.59                  ] constant for minor hwy maint estimates
3.0                     ] % increase in replace costs at end of life span for minor hwys
5                       ] yrs after repair that % increase begins for minor hwys
r                       ] reserved for future use
r                       ] reserved for future use
r                       ] reserved for future use
r                       ] reserved for future use
r                       ] reserved for future use
.
```

Note: Line numbers are provided for reference only; they do not appear in the actual parameter file.

When making modifications, avoid using tabs and do not add or remove text lines; otherwise, program results may be unreliable.

Figure 5.3 Life Cycle Model Parameter File Listing

Table 5.3 INDOT's Estimated Bridge Design Life

Steel Bridges	0 year	New
	20 years	Deck Rehabilitation
	35 years	Deck Replacement
	50 years	Deck Rehabilitation
	65 years	Bridge Replacement
Concrete Bridges	0 year	New
	20 years	Deck Rehabilitation
	35 years	Deck Rehabilitation
	50 years	Bridge Replacement

Table 5.4 Life Cycle Model Field Descriptions

Line Number	Entry	Description
1	#life_cycle_model_parameter_file	o File type identifier. o This line cannot be changed; otherwise, the program will not run.
2 - 10	Comments	o Do not remove any comment lines from the file; these lines may contain instructions or information for the user.
11	1	o Superstructure type 1 for life cycle model 1; used for internal reference only.[1]
12	Number	o Life span, in years, of bridge type 1.
13 - 16	Number Number Number Number	o Time interval, in years, between construction of bridge and rehabilitations 1 through 4. o A value of 0 indicates that a rehabilitation does not exist.
17 - 20	Number Number Number Number	o Internal codes for rehabilitations 1 through 4. o Refer to Appendix H for definition of codes.
22 - 23	m b	o Coefficients used to calculate base maintenance cost for a major highway bridge.
24	p	o Percentage increase of the replacement costs at the end of the life span for a major highway bridge.
25	c	o Number of years after a repair when maintenance costs start to increase by p for a major highway bridge.
26 - 27	m b	o Coefficients for a minor highway bridge.
28	p	o Percentage increase of the replacement costs at the end of the life span for a minor highway bridge.
29	c	o Number of years after a repair when costs begin to increase for a minor highway bridge.
30 - 36	Comments	o Do not remove.
37 - 61	see lines 11 - 29	o Definitions for superstructure type 2.

When making modifications, avoid using tabs and do not add or remove text lines; otherwise, program results may be unreliable.

Note that the equation constants are based on previous bridge records and INDOT staff assessments.

[1] Refer to Chapter 2, Section 6 for conversion of superstructure type codes from the Recording and Coding guide.

5.4.1.3 Rehabilitation Types and Times

For each bridge type, the activity profile or time interval between repairs must be determined. Furthermore, the type of rehabilitation (deck reconstruction, bridge widening, etc.) must be known. Note that the decision tree schedules immediate repairs (within six years of the analysis year), while the life cycle model predicts future repair types and times. See Table 5.3 for INDOT's estimated activity profile.

5.4.1.4 Maintenance Costs

A formula for calculating maintenance costs for each bridge type is required. After any repair, the maintenance costs remain constant for a period of time; eventually, the costs may increase linearly each year to a certain percentage of the rehabilitation costs at the end of the bridge's life span (FHWA 1987). For each superstructure type, separate cost formulas are defined for major highways and minor highways; refer to Chapter 2, Section 6 and Table 2.1 for more information regarding highway types. A typical cost model is shown in Figure 5.4. The base (or constant) maintenance cost is a linear function of the deck area and is defined as follows:

$$cost_{n_{0-c}} = (m*l*dw+b)/1000 \qquad (1)$$

where:
- $cost_n$ = thousands of dollars for year n;
- 0-c = time interval where maintenance costs are constant;

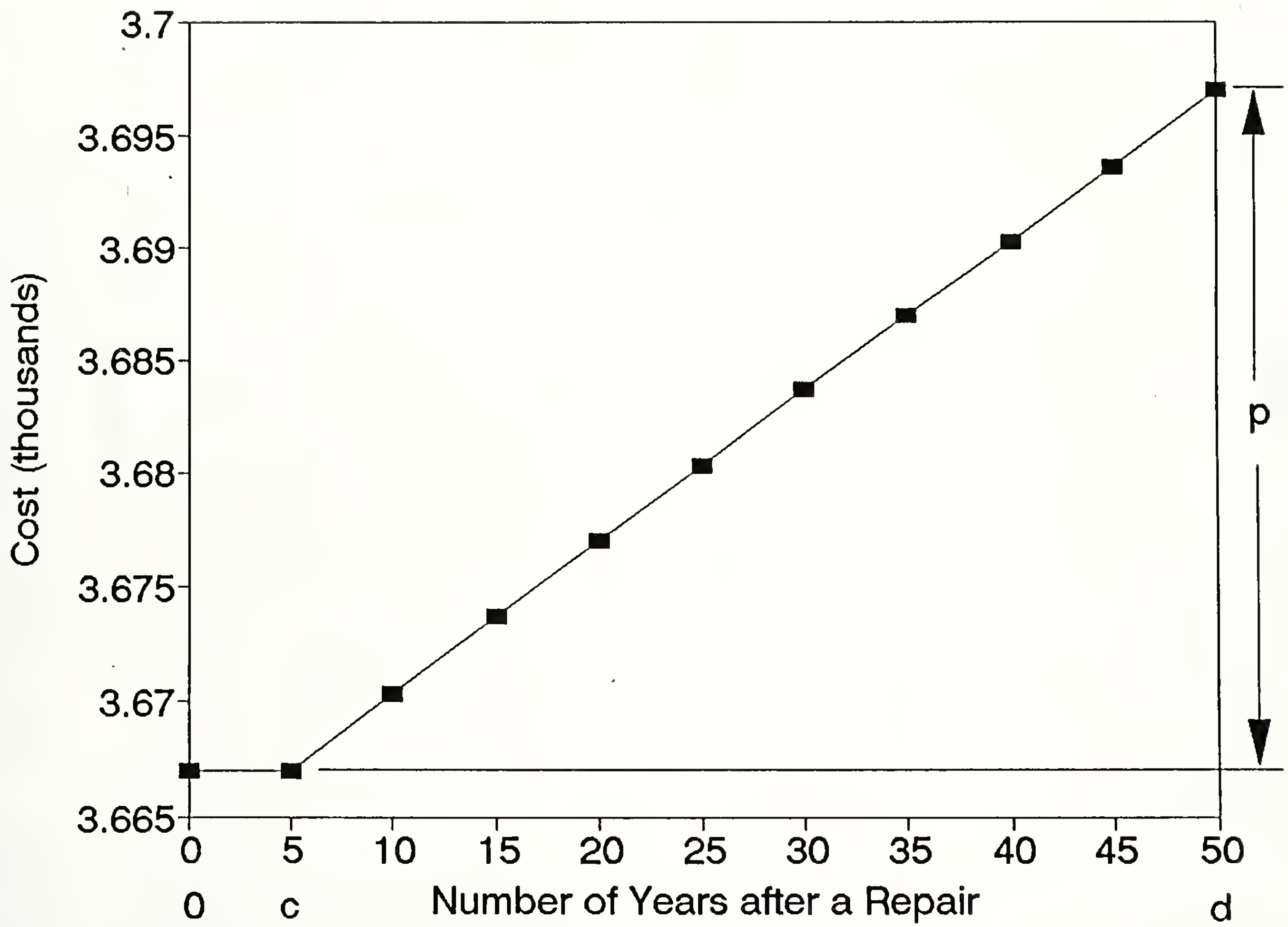


Figure 5.4 Maintenance Cost Model for Major Highways

m = slope;
l = bridge length in feet;
dw = deck width in feet;
b = constant.

Similarly, the equation representing the yearly increasing costs is written:

$$cost_{n_{c-d}} = \frac{p}{d-c} * (n-c) + cost_0 \qquad (2)$$

where:
$cost_n$ = thousands of dollars for year n;
c-d = time interval where maintenance costs are increasing;
p = percentage increase of the replacement costs at the end of the bridge's life span;
$cost_0$ = cost at year 0.

Note that these equation constants are based on previous bridge records and INDOT staff assessments.

5.5 Ranking Weight Parameter File

5.5.1 Introduction

Each bridge project is assigned a disutility or weight to determine the priority of each project. This disutility is found by calculating a series of disutility function values that rate the project based on its cost effectiveness, community impact, traffic safety, and bridge condition. All function values range from 0 to 100; as the disutility increases, the urgency for repair increases. A weighted sum of the disutilities forms the total disutility of the project. Note that this parameter file is only used

by the RANK program.

5.5.2 Specification of Weights

The parameter file is divided into three groups of weights: objective factors, bridge condition, and traffic safety. The first group computes the weighted sum (disutility) for the four criteria mentioned in the preceding paragraph. The second group combines the disutility function values calculated for remaining service life and structural condition into one disutility value, to determine the bridge condition. Similarly, the third group combines the disutility function values computed for clear deck width, vertical clearance, and inventory rating into one value, which rates the traffic safety of the bridge.

For each group, a relative set of weights is provided. These weights can be assigned any set of positive numbers; they will then be normalized so that they sum to 1.0. For example, suppose the following set of relative weights is given:

$$w1 = 50,\ w2 = 50,\ w3 = 10,\ w4 = 10$$

The sum of these four weights is obviously not 1.0, or even 100.0. Therefore, to normalize these weights, the program divides each weight by the total sum, which is:

$$50 + 50 + 10 + 10 = 120.$$

The corresponding normalized weights are then:

$$w1 = w2 = 0.416 \text{ and } w3 = w4 = 0.083.$$

Normalization is applicable for negative weights also, but should be avoided to prevent unpredictable results.

Disutility functions and weights are discussed in detail in the report, FHWA/IN/JHRP-89/12, "The Development of Optimal Strategies for Maintenance, Rehabilitation and Replacement of Highway Bridges, Final Report Vol. 5: Priority Ranking Method," Chapter 3 (Saito and Sinha 1989). The functional forms of the disutility equations are described in Section 6 of this chapter. A program listing of the file is shown in Figure 5.5; Table 5.5 provides an overview of the parameter file.

```
1  #ranking_weight_parameter_file
2  # Weighting factors used in the ranking process. All weights will be normalized
3  # to 1, so you may use any weighting system that you wish.  For instance if
4  # w1=50 w2=50 w3=10 w4=10, the weights used internally by the program would be:
5  # w1=0.416... w2=0.416... w3=0.083... w4=0.083... Do not use negative numbers!
6  c
7  c                      Weights For Objective Factors
8  10.0                   ] cost effectiveness
9  50.0                   ] bridge condition
10 30.0                   ] traffic safety
11 10.0                   ] community impact
12 c
13 c                      Two of the weights are broken down further
14 c
15 c                      Weights For Bridge Condition
16 60.0                   ] remaining service life
17 40.0                   ] structural condition
18 c
19 c                      Weights For Traffic Safety
20 30.0                   ] clear deck width
21 30.0                   ] vertical clearance
22 40.0                   ] inventory rating
```

Note: Line numbers are provided for reference only; they do not appear in the actual parameter file.

When making modifications, avoid using tabs and do not add or remove text lines; otherwise, program results may be unreliable.

Figure 5.5 Ranking Weight Parameter File Listing

Table 5.5 Ranking Weight Field Descriptions

Line Number	Entry	Description
1	#ranking_weight_parameter_file	o File type identifier. o This line cannot be changed; otherwise, the program will not run.
2 - 7	Comments	o Do not remove any comment lines from the file; these lines may contain instructions or information for the user.
8 - 11	Weight Weight Weight Weight	o Weights for objective factors; these four weights comprise the overall rank. o Note that a rank of 100 is written as 99 in the reports to save space.
16 - 17	Weight Weight	o Weights for bridge condition.
20 - 22	Weight Weight Weight	o Weights for traffic safety.

5.6 Ranking Utility Parameter File

The disutility functions are made up of a series of equations and if-then statements, which calculate the disutility, or measure of the bridge's condition, for several factors of a bridge project. These factors include economic effectiveness (E), remaining service life (R), structural condition (S), clear deck width (C), vertical clearance (V), inventory rating (I), and detour length (D). Note that this parameter file is only used in the RANK program.

Once the disutilities are determined for each aspect, they are combined and weighted to form a total disutility of the bridge project; refer to Chapter 5, Section 5. As the disutility increases, the urgency for repair increases. For a more detailed discussion of disutility functions, see the report, FHWA/IN/JHRP-89/12, "The Development of Optimal Strategies for Maintenance, Rehabilitation and Replacement of Highway Bridges, Final Report Vol. 5: Priority Ranking Method," Chapter 3 (Saito and Sinha 1989). A program listing of the file is included in Figure 5.6; Table 5.6 provides an overview of the parameter file.

```
#ranking_utility_parameter_file
#
#
#
#
c
c                        Economic Effectiveness Utility Function (E)
c                           adte = 365*adt/cef
c                           std  = standard deviation of adte for all bridges
c                           mean = average adte for all bridges
c
3.0                      ] a1: if (adte <= mean - a1*std ) then
50.0                     ] a2:       E = 100.0
c                              else if (adte >= mean + a1*std) then
c                                    E = 0.0
c                              else
c                                    E = a2*[a1*std - (adte-mean)]/a1/std
c
c
c                        Remaining Service Life Utility Function (R)
c                           ers = estimated remaining service life (yrs)
c
5.0                      ] b1: if (ers <= b1) then
15.0                     ] b2:      R = 100.0
150.0                    ] b3: else if (ers > b1 and ers < b2) then
10.0                     ] b4:      R = b3 - b4*ers
c                              else
c                                   R = 0.0
c
c
c                        Structural Condition Utility Function (S)
c                           scr = minimum of deck, superstr., or substr. condition ratings
c
3.0                      ] c1: if (scr <= c1) then
8.0                      ] c2:      S = 100.0
160.0                    ] c3: else if (scr > c1 and scr < c2) then
20.0                     ] c4:      S = c3 - c4*scr
c                              else
c                                   S = 0.0
c
c
c                        Clear Deck Width Utility Function (C)
c                           sddw = 12*(number of lanes) + 2 = standard deck width
c                           cdw  = clear deck width (ft)
c                           diff = cdw - sddw
c
-2.0                     ] d1: if (diff <= d1) then
10.0                     ] d2:      C = 100.0
c                              else if (diff >= d2) then
c                                   C = 0.0
c                              else
c                                   C = 100 * (diff-ud2)/(ud1-ud2)
c
c                        Vertical Clearance Utility Function (V)
c                           vc = vertical clearance (ft)
c                         e1 through e5 are used for major hwys
c                         e6 through e10 are used for minor hwys
14.0                     ] e1: if (vc <= e1) then
16.0                     ] e2:      V = 100.0
1600.0                   ] e3: else if (vc > e1 and vc <= e2) then
100.0                    ] e4:      V = (e3 - e4*vc) / e5
2.0                      ] e5: else
12.0                     ] e6:      V = 0.0
14.0                     ] e7:
1400.0                   ] e8:
100.0                    ] e9:
2.0                      ] e10:
                         c
```

Figure 5.6 Ranking Utility Parameter File Listing

```
69  c
70  c
71  c                         Inventory Rating Utility Function (I)
72  c                            ir = inventory rating (tons)
73  c
74  5.0                       ] f1: if (ir <= f1) then
75  25.0                      ] f2:      I = 100.0
76  125.0                     ] f3: else if (ir > f1 and ir < f2) then
77  5.0                       ] f4:      I = f3 - f4*ir
78  c                               else
79  c                                    I = 0.0
80  c
81  c
82  c                         Detour Length Utility Function (D)
83  c                            d = detour length (mi)
84  c
85  5.0                       ] g1: if (d <= g1) then
86  50.0                      ] g2:      D = 0.0
87  20.0                      ] g3: else if (d > g1 and d <= g2) then
88  100.0                     ] g4:      D = (g3*d - g4) / g5
89  9.0                       ] g5: else
90  c                                    D = 100.0
```

Note: Line numbers are for reference only; they do not appear in the actual parameter file.

Figure 5.6, continued

Table 5.6 Ranking Utility Field Descriptions

Line Number	Entry	Description
1	#ranking_utility_parameter_file	o File type identifier. o This line cannot be changed; otherwise, the program will not run.
2 - 11	Comments	o Do not remove any comment lines from the file; these lines may contain instructions or information for the user.
12 - 13	a1 a2	o Determine E by comparing average daily traffic effectiveness with the mean and standard deviation of all bridges.
23 - 26	b1...b4	o Determine R; depends on life cycle model and repair suggested by DTREE.
34 - 37	c1...c4	o Determine S; based on condition rating of superstructure, substructure, and deck.
47 - 49	d1...d3	o Determine C by comparing clear deck width to a standard width.
56 - 60	e1...e10	o Determine V by comparing standard of 16′ and 14′ vertical clearance for major and minor highways, respectively.[1]
67 - 70	f1...f4	o Determine I.
78 - 82	g1...g5	o Determine D.

[1] Vertical clearance for bridges over water are assigned a default value or specified in an exception file; see Chapter 6.

5.7 Dollar Conversion Factor Parameter File

All cost estimates made by the IBMS are in dollar values corresponding to a particular year, which will be referred to as the base year. In the current version of the IBMS, all cost estimates are in 1985 dollars. Therefore, to make the reports meaningful, estimates are converted to the year of analysis, using the dollar conversion factor parameter file.

The dollar conversion parameter file allows the user to specify up to 50 years, along with their corresponding conversion factors. Since future values of the dollar are unknown, the parameter file allows new factors to be added when available. To add new factors, simply follow the format of the existing file. A program listing of the file is shown in Figure 5.7; Table 5.7 provides an overview of the parameter file.

Costs can be converted from one year to another by taking a ratio of the respective conversion factors, which are derived from FHWA's "Price Trends for Federal-Aid Highway Construction, Third Quarter, 1991." For example, to convert a 1985 cost of $1000 to 1988 dollars, the following formula is used:

$$1988\ cost = (1985\ cost) * (1988\ factor) / (1985\ factor) \quad \textbf{(3)}$$

Therefore, 1988 Cost = ($1000) * 106.6 / 102.0 = $1045.1

To convert costs beyond 1991, the following polynomial is used:

$$a+b(y-y_o)+c(y-y_o)^2+d(y-y_o)^3+e(y-y_o)^4 \tag{4}$$

where:
- a,b,c,d,e = coefficients specified in parameter file
- y = year of analysis
- y_o = coefficient specified in parameter file

This polynomial fit was based on data from 1988 to 1991; these values can be modified.

```
#dollar_conversion_factors
# Factors used to convert dollar values from one year to another.
# You may append years to this list, up to a total of 50.
# Numbers are from FHWA PRICE TRENDS for FEDERAL-AID HIGHWAY CONSTRUCTION,
# 3rd quarter 1991.
.                         Polynomial for extending range beyond end of list
.                         a + b(y-yo) + c(y-yo)**2 + d(y-yo)**3 + e(y-yo)**4
0.0                       ] yo (use a real number here)
-290.45                   ] a  Equation is based on a fit of the data for
0.2                       ] b  1988, 1989, 1990 and 1991.
0.0                       ] c
0.0                       ] d
0.0                       ] e
c
8.0                       Annual interest rate for cost calculations
c
1985                      ] base year to convert to and from
102.0                     ] coefficient for base year
c
1972                      ] first year of list
38.6                      ] coefficient for first year
c
1973                      ] second year of list
42.5                      ] coefficient for second year
c
1974                      ] etc...
57.9                      ]
c
1975                      ]
58.1                      ]
c
1976                      ]
56.3                      ]
c
1977                      ]
59.8                      ]
c
1978                      ]
70.7                      ]
c
1979                      ]
85.5                      ]
c
1980                      ]
97.2                      ]
c
1981                      ]
94.2                      ]
c
1982                      ]
88.5                      ]
c
1983                      ]
87.6                      ]
c
1984                      ]
92.6                      ]
c
1985                      ]
102.0                     ]
c
1986                      ]
101.1                     ]
c
1987                      ]
100.0                     ]
```

Figure 5.7 Lollar Conversion Factor Parameter File Listing

```
67  c
68  1988              ]
69  106.6             ]
70  c
71  1989              ]
72  107.7             ]
73   c
74  1990              ]
75  108.5             ]
76   c
77  1991              ]
78  107.0             ]
```

Note: Line numbers are provided for reference only; they do not appear in the actual parameter file.

When making modifications, avoid using tabs and do not add or remove text lines; otherwise, program results may be unreliable.

Figure 5.7, continued

Table 5.7 Dollar Conversion Field Descriptions

Line Number	Entry	Description
1	#dollar_conversion_factors	o File type identifier. o This line cannot be changed; otherwise, the program will not run.
2 - 7	Comments	o Do not remove any comment lines from the file; these lines may contain instructions or information for the user.
9 - 13	yo a b c d e	o Values for the polynomial used to convert to and from years beyond the last year listed.[1] o Conversion factor is calculated from year y, which is known. o a - e may be set to 0.
15	Number	o Annual percentage rate used in present worth calculations. o Written as 8.0 or 5.0, not .08 or .05.
17 - 18	Base year Conversion factor	o Cost estimates are calculated in base year dollars.
20 - 21	Year Conversion Factor	o First year in file and its corresponding conversion factor.
23 - 24	Year Conversion Factor	o Second year in file and its corresponding conversion factor.
n - n+1	Year Conversion Factor	o Last year in file and its corresponding conversion factor.

[1] This polynomial fit was based on data from 1988 to 1991.

CHAPTER 6 EXCEPTION FILES

6.1 Introduction

Exception files enable the user to control how programs in the IBMS use data found in the bridge database. These files are not essential for normal operation of the system; they merely provide additional flexibility and control. Exception files allow the user to modify information in the bridge database, and thus, override decisions made by DTREE and influence how certain actions are performed, such as bridge widening. Each bridge in the database may have its own exception record. To use these files, set the "Process Exceptions" option to "y" in the runfile; refer to Chapter 4 for more details.

An exception record consists of a bridge key and several data fields. The bridge key (positions 1 through 9) contains a 5-character bridge number and 4-character bridge designation. Note that these characters may include blank spaces. This key must match the first 9 characters of the bridge record in the database file that the exception record corresponds to. An exception record can be created by modifying an existing exception file, found in the directory

\bridge\examples, using the exception editor described in Chapter 7, Section 2.9. Be sure to place the modified file in the directory \bridge\run and that the actual exception file name and the name found in the runfile match.

6.2 File Format

An exception file consists of one or more exception records. Each record consists of 36 characters and is terminated by a "new line" character. Records in the file must be sorted by the bridge key, in increasing order; otherwise, the records may not be read. The exception file may be sorted with the program EXPSORT.EXE, found in the IBMS Tools; see Chapter 7.2.6 for more information. In addition, the searching algorithm which locates the exception records requires that an exception file contain no more than approximately 32,000 records.

6.3 Record Layout and Description

The format for each record is listed in Table 6.1. For the record to be valid, each exception field must contain at least one value and appear in the proper position. However, all fields, except for the bridge key, may be set to a value which will cause them to be ignored; see Table 6.2 for a summary of these values. Sample records can be found in Appendix F. Exception files can specify six functions: repair

action, repair year, bridge length, clear deck width, approach length, and vertical clearance.

Table 6.1 Exception Record Format

Field Name	Type	Format	Position of Record (spaces)
Bridge Key	alphanumeric	a9	1-9
Exception Action	integer	i2	10-11
Exception Year	integer	i4	12-15
Exception Bridge Length	integer	i6	16-21
Exception Bridge Width	real	f5.1	22-26
Exception Approach Length	integer	i6	27-32
Exception Vertical Clearance (ft)	integer	i2	33-34
Exception Vertical Clearance (in)	integer	i2	35-36

Note that each exception field must contain at least one value; otherwise, unexpected errors or results may occur. When the positions for each field are not completely filled, values should be right-justified; examples are shown in Section 6.5

Table 6.2 Valid Values for Exception Fields

Field Name	Value which ignores exception field	Value which overrides IBMS decisions or original data
Action	< 0	0 through 17
Year[1]	< = 0	4-digit number
Bridge Length	< = 0	> 0
Bridge Width	< = 0	> 0
Approach Length	< = 0	> 0
Vertical Clearance (ft)	< = 0	> 0
Vertical Clearance (in)	< = 0	> 0

[1] A value less than the current year will still be processed by DTREE but may yield unreliable or unreasonable results.

6.3.1 Bridge Key

This field is a 9-character designation shared by all programs and files in the system. The key in the exception record must match the key in the bridge record that the exception record corresponds to. It is possible for exception records to exist with keys that do not match any bridge record in the database file being analyzed; however these exceptions will not affect program operation in any way.

6.3.2 Exception Action

Under normal system operation, the module DTREE reads the bridge database file and determines which action (replace, rehabilitation, or do nothing) to take. However, the "exception action" field may be set to override any decision made by DTREE. This feature is useful when the repair action has already been determined. For instance, if a bridge will never be replaced, only rehabilitated, an exception record can indicate this decision, and thus avoid overestimating costs. To indicate when and what action is performed on a bridge, an exception record can specify both the repair action and year (see Section 6.3.3 of this chapter).

This field may be set to a 2-digit integer code of the action desired, with possible values ranging from 0 to 17; see Appendix H or Figure 2.2 for an explanation oı these codes. A value greater or equal to zero and less than 18 instructs DTREE to use that code as the recommended action for the

bridge; hence, DTREE will not proceed with the normal decision-making process. Conversely, a value less than zero will cause the field to be ignored.

6.3.3 Exception Year

Normally, DTREE will recommend the year for which any action will be performed; however, a 4-digit year can be specified to override any year recommended by DTREE. Note that the project action will remain the same, unless the user specifies an "exception action." If a value less than zero is assigned, this field is ignored, and DTREE will proceed as normal and recommend an action year. Note that a value less than the current year will still be processed by DTREE, but may yield unreliable or unreasonable results.

6.3.4 Exception Length

If the "Lengthen Bridges with exceptions and replace?" option is set to "y" in the runfile, the bridge length will change whenever a decision specifies a bridge replacement. Conversely, to leave all bridge lengths unchanged, set this

option to "n". This 6-digit field can be set in one of three ways; if the value is:

a) = 0, the bridge is lengthened according to the curves found in the Recording and Coding Guide, Item 76.

b) < 0, the length will remain unchanged, regardless of how options are set in the runfile.

c) > 0, the length will be set to the value specified in the exception record.

Note that a bridge may be lengthened only once for all life cycle calculations.

6.3.5 Exception Width

If the "Widen Bridges with exceptions and rehab?" option is set to "y" in the runfile, the bridge width will change whenever an action specifies bridge widening. Similarly, if this option is set to "n", all bridge widths remain unchanged. Like bridge lengthening, widening can only occur once for all life cycle calculations.

The options for this field are similar to those of the "exception length," in Section 6.3.4 of this chapter. This 6-digit field can be set in one of three ways; if the value is:

a) = 0, the bridge is widened according to an internal subroutine called "WidenBridge," found in the program EXTEND.FOR.

b) < 0, the width will remain unchanged, regardless of how options are set in the runfile.

c) > 0, the width will be set to the value specified in the exception record.

6.3.6 Exception Approach Length

Bridge approach length is used to estimate the costs of certain actions; however, this value is not available in the main bridge inventory database, since it is an estimated length of approach work for a proposed bridge project. Therefore, the IBMS assigns a default value of 100 feet for all bridges. However, if an exception record specifies a positive non-zero approach length, this value will replace the default value. Conversely, a value less than or equal to zero causes this field to be ignored; thus, the default value of 100 feet will be used.

6.3.7 Exception Vertical Clearance

Although vertical clearance is specified in the bridge database, this field has a value of zero for bridges over water. In these cases, the IBMS assigns a default value of 16 feet for the vertical clearance, for both major and minor highways. However, if an exception record specifies a positive non-zero vertical clearance, this value replaces the vertical clearance in the bridge record, regardless of the value specified in the bridge record. If a value less than or equal to zero is specified, this field is ignored.

6.4 Using Exception Files

The user may specify two types of exception files:

primary and secondary exception files. The primary exception file contains a set of exception records which may change occasionally, while the secondary exception file contains records that change frequently.

During program execution, both exception files are opened. When a bridge record is read from the database, the program searches both exception files for the bridge key in the database. Certain rules are followed during this search:

1) If no exception record is found in either file, processing continues as usual.

2) If an exception record is found in only one of the exception files, primary or secondary, this file is used as described in Section 6.3 of this chapter.

3) If an exception record is found in both files, information in the secondary exception file takes precedence. However, if any fields in the secondary file are set to values that would cause these fields to be ignored, then values in those same fields in the primary file are used.

This method establishes a semi-permanent database of exception records, found in the primary exception file. Thus, if the user wishes to modify the primary file, one can easily create a secondary exception file with the new information.

As described earlier, the primary exception file contains records which do not change frequently. Consequently, exception records for approach length and vertical clearance should be placed in this file, since they rarely change, if at all. Similarly, for bridges which almost always have the same repair performed, include an "exception action" in this file. Any other information known to be relatively constant should

be included in the primary file. Typically, this exception file will be created once, with minor modifications made when necessary.

The secondary exception file contains records which often change. This file is typically smaller than the primary file, which may contain thousands of records, and used on a short term basis. Consequently, the secondary file can be easily modified when changes are desired for the current bridge analysis.

Both exception files, primary and secondary, can be controlled by setting options in the runfile. To disable both exception files, set the "Process Exceptions" option in the runfile to "n". Similarly, to disable the second exception file only, set the "Use Second Exception File" option in the runfile to "n".

All exception files must be named; default file names have been provided in the runfile for the primary and secondary exception files: EXCEPT.EXP and EXCEPT2.EXP, respectively. Each time an exception record is used, a flag is set in the reports file to indicate that an exception has been processed.

6.5 Examples

Several examples of exception files are listed below. Note that each file conforms to the standards discussed previously in Section 6.3 of this chapter. Sample files can

be found in the directory \bridge\examples. Once exception files have been created, they can be edited using the program EXEDIT.EXE; see Chapter 7 for more information.

6.5.1 Case 1

Bridge 04236ABCD will not be replaced in the near future; however, its condition may cause DTREE to suggest a replacement, producing overestimated cost calculations. To avoid this situation, prepare an exception record with the following information:

> Bridge Key - 04236ABCD
> Exception Action - code indicating a rehabilitation, say 1, Deck Rehabilitation, for example.

Other codes are set to values which cause them to be ignored; hence, the exception record would appear as follows:

```
1   5    10   15   20   25   30   35   40
----+----+----+----+----+----+----+----+
04236ABCD 1   0     0  0.0     0 0 0
```

This record is placed in a primary exception file so that this exception information will remain in the database upon each program execution; use of this record will be flagged in all reports to remind the user.

6.5.2 Case 2

The user knows that bridge 98765ZYXW will be replaced in 2000. An exception record is prepared with the following information:

Bridge Key - 98765ZYXW
Exception Action - 16, the IBMS code for replace
Exception Year - 2000.

Other fields in the exception record are set to values which cause them to be ignored; hence, the exception record would be written:

```
1   5    10   15   20   25   30   35   40
----+----+----+----+----+----+----+----+
98765ZYXW162000     0  0.0      0 0 0
```

If the "Process Exceptions" options is set to "y" in the runfile, DTREE will use this information as the repair code and year; other modules in the system will also perform calculations based on this exception record.

6.5.3 Case 3

The user plans to set the "Lengthen Bridges on Replace" in the runfile to "y", causing bridges to be lengthened when replacements are specified. The user also knows that bridges 22222CS and 88888AB will never be lengthened, bridge 33333 must be lengthened to 600 feet, and bridge 55555XX will be reduced to 75 feet.

In this case, records will be created in both the primary and secondary exception files. Since the user knows that the lengths will never change for bridges 22222CS and 88888AB, two separate records are created in the primary file, with the following information:

Exception Length = -1

Similarly, two records are created in the secondary exception file with the following information:

Exception Length = 75, for bridge 55555XX
Exception Length = 600, for bridge 33333

These records would appear as follows:

```
1   5    10   15   20   25   30   35   40
----+----+----+----+----+----+----+----+
22222CS    0   0    -1  0.0     0 0 0
88888AB    0   0    -1  0.0     0 0 0
55555XX    0   0    75  0.0     0 0 0
33333      0   0   600  0.0     0 0 0
```

6.5.4 Case 4

Bridge 7878 has an approach length of 3000 feet. If the default approach length is used, cost calculations will be underestimated. Therefore, the user prepares an exception record in the primary file, with an "exception approach length" of 3000. Since this value is placed in the primary file, the correct approach length will be used whenever the program is executed. This primary record appears in the following format:

```
1   5    10   15   20   25   30   35   40
----+----+----+----+----+----+----+----+
7878       0   0     0  0.0  3000 0 0
```

Suppose the user wants to determine the cost for a different approach length, say 2000 feet. Leaving the primary exception file intact, an exception record for 7878 is created in a secondary exception file called EXGUESS.EXP, for example,

with an "exception approach length" of 2000. The secondary record is similar to the primary, with the following modification:

```
1   5   10   15   20   25   30   35   40
----+----+----+----+----+----+----+----+
7878       0    0      0  0.0  2000 0 0
```

The option "Use Second Exception File" is set to "y" in the runfile and the file name EXGUESS.EXP is entered at the appropriate place in the runfile. Upon execution, the program finds both exception records for 7878; however, as described previously in Section 4 of this chapter, the secondary file overrides any information in the primary file. Therefore, the approach length of 2000 is used, rather than 3000.

6.5.5 Case 5

Bridge 99999Z is a bridge over water; consequently, a vertical clearance of 0 is assigned in the database record for this bridge. In reality, the bridge is known to be 95 feet above the water. If the default vertical clearance is used, substructure costs will be underestimated. To avoid this problem, the user adds a record to the primary exception file and sets the "exception vertical clearance" to 95, so that the correct vertical clearance is used upon each program execution; the record would be written:

```
1   5   10   15   20   25   30   35   40
----+----+----+----+----+----+----+----+
99999Z     0    0      0  0.0       095 0
```

CHAPTER 7 IBMS TOOLS

Several tools are available to make the IBMS more convenient to use. The IBMS Tools include the following:

1. Print and/or check bridge database file
2. Print and/or check exception file
3. Format .txt file for printer
4. Sort a Bridge Database File (.dat)
5. Sort an Exception File (.exp)
6. Sort a Text Report (.txt)
7. Generate a Year-Sum Report
8. Split a .txt file into head, body, and tail
9. Edit an Exception File
10. View Reports

7.1 Setting Up the IBMS Tools

The installation process described in Chapter 3 lists the files needed to execute these tools; however, the user must set up the IBMS Tools program group in the OS/2 Window, which was described previously in Chapter 3. The files for the IBMS tools are found on IBMS Disk#2 and a listing of these programs and their location are given in Table 7.1. Note that these instructions were written specifically for version 2.0 of OS/2; however, set-up procedures for other versions should be similar. For more information about installing programs, refer to "IBM Operating System/2 Standard Edition Version 2.0 Getting Started," Chapter 3.

Table 7.1 IBMS Tools Set-Up Procedure

Program title	Path and file name	Parameters[1]	Working directory	Program type
Print and/or check bridge database file	\bridge\prog\prbridge.cmd		\bridge\bridge	default
Print and/or check exception file	\bridge\prog\printexe.cmd		\bridge\run	default
Format .txt file for printer	\bridge\prog\prformat.cmd		\bridge\reports	default
Sort a Text Report (.txt)	\bridge\prog\textsort.cmd		\bridge\reports	default
Sort a Bridge Database File (.dat)	\bridge\prog\datasort.cmd		\bridge\bridge	default
Sort an Exception File (.exp)	\bridge\prog\expsort.cmd		\bridge\run	default
Generate a Year-Sum Report	\bridge\prog\costsum.cmd		\bridge\reports	default
Split a .txt file into head, body, and tail	\bridge\prog\split.cmd		\bridge\reports	default
Edit an Exception File	\bridge\prog\exedit.exe		\bridge\run	default
View Reports	\bridge\prog\list.exe	[Enter file name and extension]	\bridge\reports	default

[1] No information is needed for this field, except as noted.

7.2 Running the IBMS Tools from the Icon View

7.2.1 Print and/or Check Bridge Database File

This tool (PRBRIDGE.CMD) separates the fields in a bridge database file, allowing the user to easily view or print the contents of the file.

After the program is selected, all files with the .dat extension in the directory \bridge\bridge are listed. The output file name can be any eight-letter name, with a three-letter extension. Typically, the name of the bridge file is used, along with the .txt extension. If the "map records" option is selected, the program will convert codes from the Recording and Coding Guide to codes used by the IBMS; see Chapter 2, Section 6. The next option checks the file for errors; if selected, a list of all errors will be printed at the end of the file. If the map and check records options are selected, the words "Mapped" and "Checked" will appear at the top of the new file.

After the program is executed, if the .txt extension is specified, the output file will be moved to the directory \bridge\reports; otherwise, the file will remain in the \bridge\bridge directory.

7.2.2 Print and/or Check Exception File

This tool (PRINTEXE.CMD) is similar to the one described in the preceding paragraph, only exception files are formatted

for printing.

After the program is selected, a list of all .exp files in the directory \bridge\run will be generated. Again, the output file name can be any eight-letter name with a three-letter extension, but typically, the name of the exception file will be used, with the .txt extension. The next option checks for errors, such as duplicate records and whether or not the records are sorted by bridge number. These errors will be listed at the end of the file. If this option is selected, the word "Checked" will appear at the top of the new file.

After the program is executed, if the .txt extension is specified, the file will be moved to the \bridge\reports directory; otherwise, it will remain in the \bridge\run directory.

7.2.3 Format .txt File for Printer

This tool (PRFORMAT.CMD) formats a report to be printed on an 8-1/2" by 11" page.

After the program is selected, a list of all .txt files in the directory \bridge\reports is provided. The output file can be any eight-letter name and three-letter extension; however, do not specify a .txt extension or the original file will be overwritten. The file will be formatted for 65 lines per page unless specified otherwise. After execution, the file can be found in the reports directory. The formatted

file can then be printed with a text editor, such as the OS/2 System Editor, or a word processing program.

7.2.4 Sort a Text Report (.txt)

This tool (TEXTSORT.CMD) sorts a report according to different parameters, such as bridge number, cost, rank, etc.

After the program is selected, a list of all .txt files in the \bridge\reports directory is provided. The report can be sorted by two sort keys: primary and secondary. The available parameters are listed below; "N/A" means that the parameter is not available and "Nothing" tells the program not to sort the file.

N/A: Route Number	8: EUAC
2: Bridge Number	9: CEF
3: District	10: Rank
4: County	11: Rank After
5: Action Year	12: Delta Rank
6: Action Code	13: Nothing
7: Project Cost	

The file is sorted and saved within the file specified in Step 3 and can be found in the \bridge\reports directory.

7.2.5 Sort a Bridge Database File (.dat)

This tool (DATASORT.CMD) is similar to the one described in the preceding section, only it sorts a bridge database file.

After the program is selected, a list of all .dat files in the \bridge\bridge directory is provided. The report can be sorted by two sort keys: primary and secondary. The available parameters are listed below; "N/A" means that the

parameter is not available and "Nothing" tells the program not to sort the file.

1: Route Number	N/A: EUAC
2: Bridge Number	N/A: CEF
3: District	N/A: Rank
4: County	N/A: Rank After
N/A: Action Year	N/A: Delta Rank
N/A: Action Code	13: Nothing
N/A: Project Cost	

The file is sorted and saved within the file specified in Step 3 and can be found in the \bridge\bridge directory.

7.2.6 Sort an Exception File (.exp)

Exception files must be sorted by bridge number; otherwise unexpected results or errors may occur. Therefore, this tool (EXPSORT.CMD) sorts an exception file only by bridge number.

After the program is selected, a list of all .exp files in the \bridge\run directory is provided. "N/A" means that the parameter is not available and "Nothing" tells the program not to sort the file. The file is sorted and saved within the file specified in Step 2 and can be found in the \bridge\run directory.

7.2.7 Generate a Year-Sum Report

This tool (COSTSUM.CMD) computes the cumulative total each year for four parameters: cost, EUAC, CEF, and delta rank. The first three parameters calculate cumulative totals for projects selected by DTREE, COST, and RANK. The delta rank parameter applies only to RANK; it computes the cumulative total of the change in disutility before and after a repair.

After the program is selected, a list of all .txt files in the directory \bridge\reports is provided. The output file name should be different from the input file; otherwise, the input file will be overwritten. If the output file is used as an input file for another summary, the new sum will be appended to the new output file. Since a cumulative total is found for each year, the input file should be sorted by action year.

7.2.8 Split a .txt Report into Head, Body, and Tail

This tool (SPLIT.CMD) separates a text report into its three components: head, body, and tail. The head of a report contains all text above the first "cut mark" or five or more dashes; typically, the head consists of the report headers. The body of a report contains all text between the cut marks or the end of the file. The tail of a report contains the text found beneath the next cut mark or the end of the file.

After the program is selected, a list of all .txt files

in the directory \bridge\reports is provided. The new files can also be found in the directory \bridge\reports.

```
                HEAD
  ---------------------------------
                BODY
  ---------------------------------
                TAIL
```

Figure 7.1 Diagram of Head, Body, and Tail

7.2.9 Edit an Exception File

It is important that exception files follow the format described in Appendix F and Chapter 6; otherwise, unexpected results or errors may occur. This tool (EXEDIT.EXE) allows the user to edit exception files, while observing the required format. This editor consists of one main menu and four sub-menus: file, edit, move, and quit. The user may exit from any of the sub-menus by pressing the enter key; to exit from the main menu, hit "q" for quit and then "q" again or press the control and escape keys simultaneously. Instructions for using this tool to edit and save an existing file are listed below. See Table 7.2 for a description of the commands for each menu.

Table 7.2 Exception File Editor Commands

Sub-Menu	Command	Description
File	New	create a new file
	Load	load an existing file
	Save	save revisions under same file name
	Save as	save revisions under new file name
	Merge	combine two files
	Exit	quit file mode
Edit	Add	add new record
	Edit	modify current record
	Restore	undo last change
	Delete	erase current record
	Undelete	unerase last record erased
	Exit	quit edit mode
Move	Search	locate specific record by bridge number
	Up	locate previous record
	Down	locate next record
	Home	move to first record of file
	End	move to last record of file
Quit	Quit	exit editor
	Cancel	remain in editor

Exception files can be found in the \bridge\run directory. To edit a file from another directory or diskette, the full path, file name, and extension must be specified.

7.2.10 View Reports

This tool (LIST.EXE) allows the user to examine a report. The program list.exe is a read-only program that allows the user to view the contents of any ASCII file.

This program is configured to read files in the directory \bridge\reports; if the user wishes to view a file in another directory, simply type the full path name, along with the file name and extension. For example, type c:\bridge\bridge\bridge.dat.

Within the list program, the user can view another file by typing "f" and then the file name and extension. To exit, press the escape key.

7.3 Running the IBMS Tools from the OS/2 Prompt

To use the IBMS Tools from the OS/2 prompt, the user must be in the following directory: \bridge\prog. To run a tool, type the name of the file, along with the .cmd extension, then follow the instructions specified in Section 7.2 of this chapter.

The .cmd file is a batch file that provides a set of instructions for the actual program (.exe file). If the .cmd

extension is not specified, the executable file will be run directly. If this occurs, the full path must be entered when prompted for a file name. Similarly, if a path is not specified for the output file, it will be placed in the \bridge\prog directory. An example is shown in Chapter 8.

CHAPTER 8 EXAMPLE RUN OF THE IBMS

This chapter provides the step-by-step procedure for running the entire IBMS project evaluation package from the OS/2 prompt. Instrumentation for running the package from the Icon View are similar; refer to Chapter 3, Section 5. The sequential order of the results shown depend on the ordering of the bridge data base used. Individual results may differ from the example shown here.

8.1 Bridge Database File

The first step is to prepare a bridge database file, as explained in Appendix F.1, and copy it to the directory \bridge\bridge. See Table 8.1 for the bridge data base format. The file used for this example contains fifteen bridges; however, the IBMS has been tested successfully with a database of 5000 bridges. The example file is called TEST.DAT; see Figure 8.1. The database file can be "mapped," as described in Chapter 2, Section 6, and then printed with the IBMS Tools, so that the data is more meaningful. The mapping procedure is shown in Figures 8.2 and 8.3; the mapped file can be found in Figure 8.4. See Table 8.2 for a complete description of the

```
1   5    10   15   20   25   30   35   40   45   50   55   60   65   70
----+----+----+----+----+----+----+----+----+----+----+----+----+----+---
000703900652    0519281961062004200020389036900007800001020512354453 19108
002405600899B   041929199106200380002048704480000300000102061298658  9004
005607801083A   051947198506200130002034302800001620000102081316665  9008
022708901363A   0319381973074001500020374034300012621086020213257563 59108
003307102037    0419360000         00018001600001321402303007006766358912
I06504902431    03197400001110898000816051575000292220440202129557735 8912
002405202456AWBL021975199114200340002043704070001562207402011257667  9109
033105003451    041947000007400250002031302800000720000201062365666319002
I06501004216BSBL051958198901101324502042603950001241410102011328775  9005
I06900204540CSBL021961198801101260003047604130002101404402011296667349204
002405204858AWBL021963198602200700002041003800001020000201011216666  9201
I46500605289A   0319711982011      02029202620002121502202042327775  9109
025202406934    051988000007400140002046504350001010000201092368867  9104
001803807558    0219860000062001700020000000000003000003190423 6NNNN  9202
002708907710SBL 0319760000         000099006700008419035020080077 78  9104
```

Figure 8.1 Test.dat

Table 8.1 Test.dat Format

Position	Description
1-4	highway route number
5-7	county code
8-12	bridge number
13-16	bridge designation
17-18	highway dept. district code
19-22	year of original construction
23-26	last year bridge was repaired
27-28	functional class code
29	hwy system of inventory route
30-35	average daily traffic (ADT)
36-37	number of traffic lanes
38-41	total deck width (ft)
42-45	clear deck width (ft)
46-51	bridge length (ft)
52-53	vertical clearance (ft)
54-55	vertical clearance (in)
56	superstructure material
57-58	superstructure construction
59-60	detour length (mi)

Table 8.1, continued

Position	Description
61	type of load
62-63	inventory rating (tons)
64	main deck surface condition rating
65	superstructure condition rating
66	substructure condition rating
67	deck geometry code
68-69	proposed work code
70-73	date of last inspection (yr/month)

```
[A:\BRIDGE\PROG]prbridge

Print Bridge Database
---------------------

Enter Bridge File name: a:\bridge\bridge\test.dat

Enter Output File name: a:\bridge\reports\test.txt

Map Records Before Printing? (y/n) y

Check Records? (y/n) y

Successful completion, file written to a:\bridge\reports\test.txt

Hit Enter to continue...

[A:\BRIDGE\PROG]
```

Figure 8.2 Running Prbridge.exe

```
[A:\BRIDGE\PROG]prbridge.cmd
.
INDOT BMS Bridge File List and Check (PRBRIDGE)
-----------------------------------------------

.
Bridge Files in \BRIDGE\BRIDGE :

.

 Volume in drive A is BMS-OS2
 Volume Serial Number is 6574-4414
 Directory of A:\BRIDGE\BRIDGE

TEST.DAT
        1 file(s)        750 bytes
                       57856 bytes free

.
Remember the file name to be used.  If you need to see
it again, run the program over.
Press any key to continue...

.

Print Bridge Database
---------------------

Enter Bridge File name: test.dat

Enter Output File name: test.txt

Map Records Before Printing? (y/n) y

Check Records? (y/n) y

Successful completion, file written to test.txt


Hit Enter to continue...

.
Moving .txt report generated to \BRIDGE\REPORTS ...

.
TEST.TXT
         1 file(s) copied

.
Press any key to continue...

[A:\BRIDGE\PROG]
```

Figure 8.3 Running Prbridge.cmd

Mapped — INDOT BMS Bridge database file: test.dat — Checked

Route	Co.	Br # des	Dis	YrBlt	Fixed	FC	Hwy	ADT	Lns	DkWid	ClWd	Length	Vft	Vin	SM	SC	Det	IR	Ton	DC	SC	sC	DG	WK	IDate	DeckT
0007	39	00652	5	1961	1961	2	2	4200	2	38.9	36.9	78	16	0	2	4	5	1	23	5	4	4	5	31	9108	*
0024	56	00899B	4	1991	1991	2	2	3800	2	48.7	44.8	30	16	0	2	4	6	1	29	8	6	5	8	0	9004	*
0056	78	01083A	5	1985	1985	2	2	1300	2	34.3	28.0	162	16	0	2	4	8	1	31	6	6	6	5	0	9008	*
0227	89	01363A	3	1973	1973	1	4	1500	2	37.4	34.3	126	21	8	2	2	2	1	32	5	7	5	6	35	9108	*
0033	71	02037	4	1936	0	1	0	0	0	18.0	16.0	132	14	2	1	4	0	7	0	6	7	6	6	35	8912	*
1065	49	02431	3	1974	0	1	1	89800	8	160.5	157.5	292	22	4	1	4	2	1	29	5	5	7	7	35	8912	*
0024	52	02456AWBL	2	1991	1991	1	2	3400	2	43.7	40.7	156	22	7	1	4	1	1	25	7	6	6	7	0	9109	*
0331	50	03451	4	1947	0	1	4	2500	2	31.3	28.0	72	16	0	2	1	6	2	36	5	6	6	6	31	9002	*
1069	2	04540CSBL	2	1988	1988	1	1	12600	3	47.6	41.3	210	14	4	1	4	1	1	29	6	6	6	7	34	9204	*
0024	52	04858AWBL	2	1986	1986	1	2	7000	2	41.0	38.0	102	16	0	2	1	1	1	21	6	6	6	6	0	9201	*
1465	6	05289A	3	1982	1982	1	1	0	2	29.2	26.2	212	15	2	2	4	4	2	32	7	7	7	5	0	9109	*
0252	24	06934	5	1988	0	1	4	1400	2	46.5	43.5	101	16	0	2	1	9	2	36	8	8	6	7	0	9104	*
0018	38	07558	2	1986	0	2	2	1700	2	0.0	0.0	30	16	0	1	2	4	2	36	*	*	*	*	0	9202	*
0027	89	07710SBL	3	1976	0	1	0	0	0	9.9	6.7	84	19	3	2	2	0	8	0	7	7	7	8	0	9104	*
1065	10	04216BSBL	5	1989	1989	1	1	13245	2	42.6	39.5	124	14	10	2	4	1	1	32	8	7	7	5	0	9005	*

```
Errors in bridge record 02037      1 fatal and 2 nonfatal.
      HwyType   nonfatal
      ADT       fatal
      NumLanes  nonfatal
Errors in bridge record 05289A     1 fatal and 0 nonfatal.
      ADT       fatal
Errors in bridge record 07558      6 fatal and 0 nonfatal.
      DkWidth   fatal
      ClDkWid   fatal
      DeckCond  fatal
      SupCond   fatal
      SubCond   fatal
      DeckGeom  fatal
Errors in bridge record 07710SBL   1 fatal and 2 nonfatal.
      HwyType   nonfatal
      ADT       fatal
      NumLanes  nonfatal
```

Figure 8.4 Mapped Bridge File

mapped bridge file format.

Note that four bridges contained fatal errors, indicating that information needed for program calculations was bad or missing. Nonfatal errors occur when data does not conform to the NBIS, but does not interfere with program operation.

8.2 Runfile

The runfile used in this example is called "test.run;" see Figure 8.5. This runfile was created by duplicating the default runfile (RUNFILE.RUN), found in the directory \bridge\run or \bridge\examples, changing the bridge database file name to TEST.DAT, and then saving the runfile under the name TEST.RUN. These modifications were made using the OS/2 System Editor. Note that the "Process Exceptions" and "Lengthen and Widen Bridges" options are set to "n." Remember that the runfile should be copied to the directory \bridge\run.

8.3 DTREE

The next step is to run DTREE, as shown in Figure 8.6. In addition, the text report, log file, output file, and Lotus-123 file are provided in Figures 8.7, 8.8, 8.9, and 8.10, respectively. Formats of the outputs of DTREE.TXT and DTREE.OUT are given in Tables 8.3 and 8.4. At this point, the IBMS simply

Table 8.2 Description of Mapped Test.dat

Heading	Description
Route	Route number
Co.	County code
Br # des	Bridge number and designation
Dis	District code
YrBlt	Year bridge was constructed
Fixed	Year of last repair
FC	Functional classification
Hwy	Highway system of the inventory route
ADT	Average daily traffic
Lns	Number of lanes
DkWid	Deck width
ClWd	Clear deck width
Length	Length of bridge
Vft	Vertical clearance, ft
Vin	Vertical clearance, in
SM	Superstructure material
SC	Superstructure construction
Det	Detour length
IR	Type of load
Ton	Inventory rating
DC	Deck condition
SC	Superstructure condition
sC	Substructure condition
DG	Deck geometry code
WK	Proposed work code
IDate	Last inspection date
DeckT[1]	Deck type

[1] This field is not used in the current IBMS.

```
#run_file
#
#
#
#
y               ] All: Print text reports? (y/n)
y               ] All: Print 123 reports? (y/n)
y               ] All: Print log file? (y/n)
y               ] All: Check bridge records? (y/n)
n               ] All: Process exceptions? (y/n)
c               ] Dtree: Report Style (n)ormal (c)ondition (e)xception
n               ] Dtree: Use second exception file? (y/n)
                ] Dtree:
                ] Dtree:
                ] Dtree:
l               ] Cost: Style (n)orm pres(w)orth (l)ookahead (p)wlookahead
y               ] Cost: Do maintenance cost calculations? (y/n)
n               ] Cost: Lengthen Bridges with exceptions and replace? (y/n)
n               ] Cost: Widen Bridges with exceptions and rehab? (y/n)
                ] Cost:
u               ] Rank: Report style (n)ormal (u)tility
                ] Rank:
                ] Rank:
                ] Rank:
                ] Rank:
                ] Opt:
                ] Opt:
                ] Opt:
                ] Opt:
                ] Opt:
                ] reserved
                ] reserved
                ] reserved
                ] reserved
                ] reserved
c
                Bridge database files
bridge.dat      ] Bridge data file from \bridge\bridge
default         ] Bridge index file from \bridge\bridge
except.exp      ] Primary exception file
except2.exp     ] Secondary exception file
c
                Decision tree program files
dtree           ] File prefix for .out .txt .prn .log output files
dtree.par       ] Parameter file: defines decision criteria (must have .par)
c
                Economic analysis program files
dtree.out       ] Input file for cost analysis
cost            ] File prefix for .out .txt .prn .log output files
life.par        ] Life cycle model parameters (must have .par)
cost.par        ] Cost estimate coefficient parameter file
c
                Ranking program files
cost.out        ] Input file for ranking
rank            ] File prefix for .out .txt .prn .log output files
weight.par      ] Ranking weights (must have .par)
utility.par     ] Distility function parameters (must have .par)
c
                Optimization program files
rank.out        ] Input file for optimization
opt             ] File prefix for .txt .prn .log output files
opt.par         ] Optimization parameters (must have .par)
c
                Other files
dollar.par      ] Interest rate and $ conversi:.. factors (must have .par)
default         ] Log File (overrides log file name from above)
```

Figure 8.5 Test.run

```
[A:\BRIDGE\PROG]dtree
DECISION TREE PROGRAM
=====================
   ...initializing
Enter runfile name [default] a:\bridge\run\runtest.run

Enter year of analysis: 1992

Enter report style (n,c,e) [c] c

Process exceptions [n] n

   ...processing bridge file records
Errors in bridge record 02037       1 fatal and 2 nonfatal.
Errors in bridge record 05289A      1 fatal and 0 nonfatal.
Errors in bridge record 07558       6 fatal and 0 nonfatal.
Errors in bridge record 07710SBL    1 fatal and 2 nonfatal.

Total Records Successfully Processed    11
Total Errors In Processing Records       4
Decision          Count
===============   =====
Nothing               2
Deck Rehab            2
Deck Replace          1
Super&Deck Rehab      1
Sup RH Deck Repl      1
Substruct Rehab       1
Sub&Deck Rehab        1
Sub RH Deck Repl      1
Sub,Sup&Deck RH       1
Sub RH Sup Repl       2
Sub&Sup Rehab         1
Sub&Sup RH,Dk RP      1
Super Replace         1
Widen&Deck Rehab      1
Widen&Deck Repl       1
Raise/Lower Pvmt      1
Replace               1

Quitting dtree. Successful completion.
****done****
Press Enter to exit...

[A:\BRIDGE\PROG]
```

Figure 8.6 Running DTREE

RUNFILE: test.run DECISION TREE REPORT YEAR OF ANALYSIS: 1992

Bridge num des	Route #	Dis	Co. Code	Act Year	Act Code	Action	1992 Cost ($1000)	Work Code	Work Proposed	Ex?	Super Cond.	Sub Cond.	Deck Cond.	In Yr	Super @InYr	Sub @InYr	Deck @Inyr
00652	0007	5	39	1994	16	Replace	206.2	31	Replace Substd	F	3	4	5	91	4	4	5
00899B	0024	4	56	1994	10	Sub&Sup Rehab	52.3	0		F	5	4	7	90	6	5	8
01083A	0056	5	78	1994	13	Widen&Deck Rehab	185.5	0		F	5	5	5	90	6	6	6
01083A	0056	5	78	1995	14	Widen&Deck Repl	247.3	0		F	5	5	4	90	6	6	6
01363A	0227	3	89	1994	1	Deck Rehab	101.4	35	Rehab Substd	F	7	5	5	91	7	5	5
01363A	0227	3	89	1995	2	Deck Replace	282.1	35	Rehab Substd	F	7	5	4	91	7	5	5
01363A	0227	3	89	1996	7	Sub RH Deck Repl	282.1	35	Rehab Substd	F	7	4	4	91	7	5	5
02431	1065	3	49	1994	4	Sup RH Deck Repl	2945.6	35	Rehab Substd	F	5	7	4	89	5	7	5
02456AWBL	0024	2	52	1994	3	Super&Deck Rehab	154.1	0		F	4	5	6	91	6	6	7
02456AWBL	0024	2	52	1995	8	Sub,Sup&Deck RH	183.4	0		F	4	4	5	91	6	6	7
02456AWBL	0024	2	52	1997	9	Sub RH Sup Repl	383.0	0		F	3	3	4	91	6	6	7
03451	0331	4	50	1994	11	Sub&Sup RH,Dk RP	148.7	31	Replace Substd	F	4	4	3	90	6	6	5
03451	0331	4	50	1995	9	Sub RH Sup Repl	154.4	31	Replace Substd	F	3	4	3	90	6	6	5
04216BSBL	1065	5	10	1994	0	. . . . .	0.0	0		F	5	6	7	90	7	7	8
04540CSBL	1069	2	2	1994	15	Raise/Lower Pvmt	371.1	34	Widen w/ Repair	F	5	5	5	92	6	6	6
04540CSBL	1069	2	2	1996	12	Super Replace	343.6	34	Widen w/ Repair	F	4	4	4	92	6	6	6
04858AWBL	0024	2	52	1994	0	. . . . .	0.0	0		F	6	6	6	92	6	6	6
04858AWBL	0024	2	52	1996	1	Deck Rehab	136.1	0		F	6	5	5	92	6	6	6
04858AWBL	0024	2	52	1997	6	Sub&Deck Rehab	136.1	0		F	6	4	5	92	6	6	6
06934	0252	5	24	1994	5	Substruct Rehab	26.5	0		F	7	4	7	91	8	6	8

Total Records Successfully Processed 11
Total Errors In Processing Records 4

Decision	Count
Nothing	2
Deck Rehab	2
Deck Replace	1
Super&Deck Rehab	1
Sup RH Deck Repl	1
Substruct Rehab	1
Sub&Deck Rehab	1
Sub RH Deck Repl	1
Sub,Sup&Deck RH	1
Sub RH Sup Repl	2
Sub&Sup Rehab	1
Sub&Sup RH,Dk RP	1
Super Replace	1
Widen&Deck Rehab	1
Widen&Deck Repl	1
Raise/Lower Pvmt	1
Replace	1

Figure 8.7 Dtree.txt

Table 8.3 Description of Dtree.txt Condition Report

Heading	Description
Bridge num and des	Bridge number and designation
Route #	Route number
Dis	District code
Co. Code	County code
Act Year	Action year
Act Code	Action code
Action	Action
1992 Cost ($1000)	Base year cost converted to analysis year dollars
Work Code	Work code from NBIS Recording and Coding Guide
Work Proposed	Work proposed by field inspector
Ex? (T or F)	T: Exception record used F: Exception record not used
Super Cond.	Superstructure condition number in analysis year
Sub Cond.	Substructure condition number in analysis year
Deck Cond.	Deck condition number in analysis year
In Yr	Inspection year
Super@InYr, Sub@InYr, and Deck@InYr	Condition numbers in inspection year
Action Code Abbreviation[1]	
Rehab, RH	Rehabilitation
Replace, Repl, RP	Replacement
Sup, Super	Superstructure
Sub, Substruct	Substructure
Pvmt	Pavement
Dk	Deck

[1] See Appendix H for more information.

```
INDOT BMS       DECISION TREE LOG     Runfile:  test.run  YEAR OF ANALYSIS: 1992
---------------------------------------------------------------------------------

   ...processing bridge file records
 Errors in bridge record 02037      1 fatal and 2 nonfatal.
       HwyType   nonfatal
       ADT       fatal
       NumLanes  nonfatal
 Errors in bridge record 05289A     1 fatal and 0 nonfatal.
       ADT       fatal
 Errors in bridge record 07558      6 fatal and 0 nonfatal.
       DkWidth   fatal
       ClDkWid   fatal
       DeckCond  fatal
       SupCond   fatal
       SubCond   fatal
       DeckGeom  fatal
 Errors in bridge record 07710SBL   1 fatal and 2 nonfatal.
       HwyType   nonfatal
       ADT       fatal
       NumLanes  nonfatal
-----------------------------------------------------------------------------------------

 Total Records Successfully Processed    10
 Total Errors In Processing Records       0
 Decision          Count
 ===============   =====
 Nothing              1
 Deck Rehab           2
 Deck Replace         1
 Super&Deck Rehab     1
 Sup RH Deck Repl     1
 Substruct Rehab      1
 Sub&Deck Rehab       1
 Sub RH Deck Repl     1
 Sub,Sup&Deck RH      1
 Sub RH Sup Repl      2
 Sub&Sup Rehab        1
 Sub&Sup RH,Dk RP     1
 Super Replace        1
 Widen&Deck Rehab     1
 Widen&Deck Repl      1
 Raise/Lower Pvmt     1
 Replace              1
```

Figure 8.8 Dtree.log

```
1   5    10   15   20   25   30   35   40   45
----+----+----+----+----+----+----+----+----+
00652     199416345    0  0.0     0 0 0
00899B    199410547    0  0.0     0 0 0
01083A    199413555    0  0.0     0 0 0
01083A    199514554    0  0.0     0 0 0
01363A    1994 1755    0  0.0     0 0 0
01363A    1995 2754    0  0.0     0 0 0
01363A    1996 7744    0  0.0     0 0 0
02431     1994 4574    0  0.0     0 0 0
02456AWBL1994 3456     0  0.0     0 0 0
02456AWBL1995 8445     0  0.0     0 0 0
02456AWBL1997 9334     0  0.0     0 0 0
```

Figure 8.9 Dtree.out twenty projects were selected.

```
1   5    10   15   20   25   30   35   40   45
----+----+----+----+----+----+----+----+----+
03451    199411443    0  0.0     0 0 0
03451    1995 9343    0  0.0     0 0 0
04216BSBL1994 0567    0  0.0     0 0 0
04540CSBL199415555    0  0.0     0 0 0
04540CSBL199612444    0  0.0     0 0 0
04858AWBL1994 0666    0  0.0     0 0 0
04858AWBL1996 1655    0  0.0     0 0 0
04858AWBL1997 6645    0  0.0     0 0 0
06934    1994 5747    0  0.0     0 0 0
```

Figure 8.9, continued

recommends repairs for each bridge. To examine these reports, use the IBMS Tool, "View Reports."

Although only fifteen bridges appear in the database file, Since DTREE analyzes a set of bridges over a four-year period, multiple projects may be recommended for each bridge. After projects are selected, repair costs are computed, in analysis year dollars. The text report (Figure 8.7) also provides the work proposed by the field inspector, which can be compared to the project recommended by DTREE. The "condition" report lists condition numbers for the inspection year, as well as for the analysis year.

The log file (Figure 8.8) produces a record of program execution, noting any errors that may occur. Fatal errors were found in four bridges, indicating that they contained bad or missing data. Consequently, these four bridges cannot be processed by any of the programs until the errors are corrected.

The output file (Figure 8.9) provides information needed by COST. Although the information in the Lotus 123 report (Figure 8.10) appears to be out of alignment, when it is imported into a spreadsheet program, each column will be properly aligned.

8.4 COST

The procedure for running COST is shown in Figure 8.11. The corresponding output files are provided in Figures 8.12 through 8.15. At this stage, the user can select projects only on the basis of cost effectiveness.

The cost report (Figure 8.12 and Tables 8.5 and 8.6) takes the projects selected

Table 8.4 Dtree.out Format

Position	Description
1-5	bridge number in the bridge database file
6-9	bridge designation in the bridge file
10-13	action year generated by DTREE
14-15	action code generated by DTREE
16	superstructure condition at time of action
17	substructure condition at time of action
18	deck condition at time of action
19-24	exception bridge length from the exception file
25-29	exception bridge width from the exception file
30-35	exception approach length
36-37	exception vertical clearance (ft)
38-39	exception vertical clearance (in)

Note that the program dtree.for first initializes columns 19 through 39 (exception file information) with values of zero. If exceptions are used, the program will then replace the zeroes with exception information in the output file. The output file shown in Figure 8.9 did not have any exception information; consequently, several columns of zeroes appear in the file.

```
"DECISION" "TREE" "REPORT"

"RUNFILE:" "test.run    " "ANALYSIS YEAR:" 1992

"Bridge #" "Route #" "Dist." "County" "Act Yr" "A.Code" "Action"  "Cost" "Wk Code" "Wk Prop" "Exe?"          "Sup C" "Sub C" "Deck C" "InsYr"
"Sup@Yr" "Sub@Yr"             "Deck@Yr"
"---------""---------""---------""---------""---------""---------""---------""---------""---------""---------""---------""---------""---------"-
--------""---------""---------""---------""---------"
"00652    " "0007"  5  39 1994 16 "Replace           "   206.2 31 "Replace Substd  " " F" 3 4 5 91 4 4 5
"00899B   " "0024"  4  56 1994 10 "Sub&Sup Rehab     "    52.3  0 "                " " F" 5 4 7 90 6 5 8
"01083A   " "0056"  5  78 1994 13 "Widen&Deck Rehab"    185.5  0 "                " " F" 5 5 5 90 6 6 6
"01083A   " "0056"  5  78 1995 14 "Widen&Deck Repl "    247.3  0 "                " " F" 5 5 4 90 6 6 6
"01363A   " "0227"  3  89 1994  1 "Deck Rehab        "   101.4 35 "Rehab Substd    " " F" 7 5 5 91 7 5 5
"01363A   " "0227"  3  89 1995  2 "Deck Replace      "   282.1 35 "Rehab Substd    " " F" 7 5 4 91 7 5 5
"01363A   " "0227"  3  89 1996  7 "Sub RH Deck Repl"     282.1 35 "Rehab Substd    " " F" 7 4 4 91 7 5 5
"02431    " "I065"  3  49 1994  4 "Sup RH Deck Repl"    2945.6 35 "Rehab Substd    " " F" 5 7 4 89 5 7 5
"02456AWBL" "0024"  2  52 1994  3 "Super&Deck Rehab"     154.1  0 "                " " F" 4 5 6 91 6 6 7
"02456AWBL" "0024"  2  52 1995  8 "Sub,Sup&Deck RH "     183.4  0 "                " " F" 4 4 5 91 6 6 7
"02456AWBL" "0024"  2  52 1997  9 "Sub RH Sup Repl "     383.0  0 "                " " F" 3 3 4 91 6 6 7
"03451    " "0331"  4  50 1994 11 "Sub&Sup RH,Dk RP"     148.7 31 "Replace Substd  " " F" 4 4 3 90 6 6 5
"03451    " "0331"  4  50 1995  9 "Sub RH Sup Repl "     154.4 31 "Replace Substd  " " F" 3 4 3 90 6 6 5
"04216BSBL" "I065"  5  10 1994  0 "Nothing           "     0.0  0 "                " " F" 5 6 7 90 7 7 8
"04540CSBL" "I069"  2   2 1994 15 "Raise/Lower Pvmt"     371.1 34 "Widen w/ Repair " " F" 5 5 5 92 6 6 6
"04540CSBL" "I069"  2   2 1996 12 "Super Replace     "   343.6 34 "Widen w/ Repair " " F" 4 4 4 92 6 6 6
"04858AWBL" "0024"  2  52 1994  0 "Nothing           "     0.0  0 "                " " F" 6 6 6 92 6 6 6
"04858AWBL" "0024"  2  52 1996  1 "Deck Rehab        "   136.1  0 "                " " F" 6 5 5 92 6 6 6
"04858AWBL" "0024"  2  52 1997  6 "Sub&Deck Rehab  "     136.1  0 "                " " F" 6 4 5 92 6 6 6
"06934    " "0252"  5  24 1994  5 "Substruct Rehab "      26.5  0 "                " " F" 7 4 7 91 8 6 8
----------------------------------------------------------------------------------------------------
 "********" "Summary" "*********"
 "Total Records Successfully Processed"   11
 "Total Errors In Processing Records"    4
 "Decision"  "Count"
"========"  "========"
 "Nothing         "      2
 "Deck Rehab      "      2
 "Deck Replace    "      1
 "Super&Deck Rehab"      1
 "Sup RH Deck Repl"      1
 "Substruct Rehab "      1
 "Sub&Deck Rehab  "      1
 "Sub RH Deck Repl"      1
 "Sub,Sup&Deck RH "      1
```

Figure 8.10 Dtree.prn

```
"Sub RH Sup Repl "      2
"Sub&Sup Rehab   "      1
"Sub&Sup RH,Dk RP"      1
"Super Replace   "      1
"Widen&Deck Rehab"      1
"Widen&Deck Repl "      1
"Raise/Lower Pvmt"      1
"Replace         "      1
```

Figure 8.10, continued

```
[A:\BRIDGE\PROG]cost
ECONOMIC ANALYSIS PROGRAM
=========================
Initializing...
Enter runfile name [default] a:\bridge\run\runtest.run

Enter year of analysis:  1992

Enter report style (n,l,w,p) [l] l

Process Exceptions [n] n

Processing input records...
Quitting cost. Successful completion.
****done****
Press Enter to continue....

[A:\BRIDGE\PROG]
```

Figure 8.11 Running COST

RUNFILE: test.run COST PROGRAM REPORT YEAR OF ANALYSIS: 1992

Bridge num des	Route #	Dis	Co. Code	Act Year	Act Code	Action	Yr 1992 Cost ($1000)	EUAC $1000	CEF v/$/ 1000f2	Ex ?	Year	AC	Yr 1992 Cost ($1000)	Year	AC	Yr 1992 Cost ($1000)	Year	Ac	Yr 1992 Cost ($1000)
00652	0007	5	39	1994	16	Replace	206.2	20.8	25.64	F	2014	3	103.6	2029	3	103.6	2044	16	206.2
00899B	0024	4	56	1994	10	Sub&Sup Rehab	52.3	8.2	126.26	F	2009	3	49.9	2024	16	134.2	2044	3	49.9
01083A	0056	5	78	1994	13	Widen&Deck Rehab	185.5	25.0	4.19	F	2009	3	125.6	2024	16	309.1	2044	3	125.6
01083A	0056	5	78	1995	14	Widen&Deck Repl	247.3	27.1	3.86	F	2010	3	125.6	2025	16	309.1	2045	3	125.6
01363A	0227	3	89	1994	1	Deck Rehab	101.4	16.1	7.87	F	2009	3	106.5	2024	16	297.6	2044	3	106.5
01363A	0227	3	89	1995	2	Deck Replace	282.1	26.4	4.80	F	2010	3	106.5	2025	16	297.6	2045	3	106.5
01363A	0227	3	89	1996	7	Sub RH Deck Repl	282.1	24.4	5.18	F	2011	3	106.5	2026	16	297.6	2046	3	106.5
02431	1065	3	49	1994	4	Sup RH Deck Repl	2945.6	309.2	2.30	F	2009	2	2805.3	2024	3	577.3	2039	16	2019.4
02456AWBL	0024	2	52	1994	3	Super&Deck Rehab	154.1	27.6	7.08	F	2009	2	408.1	2024	3	154.1	2039	16	383.0
02456AWBL	0024	2	52	1995	8	Sub,Sup&Deck RH	183.4	27.4	7.12	F	2010	2	408.1	2025	3	154.1	2040	16	383.0
02456AWBL	0024	2	52	1997	9	Sub RH Sup Repl	383.0	34.4	5.68	F	2012	2	408.1	2027	3	154.1	2042	16	383.0
03451	0331	4	50	1994	11	Sub&Sup RH,Dk RP	148.7	16.7	27.07	F	2009	16	154.4	2029	3	77.0	2044	3	77.0
03451	0331	4	50	1995	9	Sub RH Sup Repl	154.4	15.8	28.56	F	2010	16	154.4	2030	3	77.0	2045	3	77.0
04540CSBL	1069	2	2	1994	15	Raise/Lower Pvmt	371.1	49.9	10.63	F	2009	2	598.3	2024	3	225.9	2039	16	482.5
04540CSBL	1069	2	2	1996	12	Super Replace	343.6	41.2	12.88	F	2011	2	598.3	2026	3	225.9	2041	16	482.5
04858AWBL	0024	2	52	1996	1	Deck Rehab	136.1	15.9	41.33	F	2011	3	142.9	2026	16	250.8	2046	3	142.9
04858AWBL	0024	2	52	1997	6	Sub&Deck Rehab	136.1	14.8	44.62	F	2012	3	142.9	2027	16	250.8	2047	3	142.9
06934	0252	5	24	1994	5	Substruct Rehab	26.5	10.8	10.75	F	2009	3	106.1	2024	16	280.8	2044	3	106.1

===== Summary ====
Records Processed: 18
Total Errors : 0

Total Cost ($1000) 6339.3
Total EUAC ($1000) 711.7
Total CEF ($1000) 375.8

AVE Cost ($1000) 352.2
AVE EUAC ($1000) 39.5
AVE CEF ($1000) 20.9

Figure 8.12 Cost.txt

Table 8.5 Description of COST Look Ahead Report

Heading	Description
Bridge num and des	Bridge number and designation
Route #	Route number
Dis	District code
Co. Code	County code
Act Year	Action year
Act Code	Action code
Action	Action
1992 Cost ($1000)	Base year cost converted to analysis year dollars, in thousands
EUAC $1000	Equivalent uniform annual cost, in analysis year dollars
CEF v/$/1000f2	Cost effectiveness factor, in vehicles/$1000/feet^2
Ex? (T or F)	T: Exception record used F: Exception record not used
Year	Year of next repair in life cycle
Ac	Repair code
Yr 1992 Cost ($1000)	Estimated cost, in analysis year dollars

```
RUNFILE: runfile.run    COST    PROGRAM    LOG         YEAR OF ANALYSIS: 1992

-------------------------------------------------------------------------------
 Processing input records...
-------------------------------------------------------------------------------

===== Summary ====
Records Processed:    18
Total Errors     :    0

Total Cost ($1000)    6339.3
Total EUAC ($1000)     711.7
Total CEF  ($1000)     375.8

AVE   Cost ($1000)     352.2
AVE   EUAC ($1000)      39.5
AVE   CEF  ($1000)      20.9
```

Figure 8.13 Cost.log

```
1   5    10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+
00652     199416345    0  0.0     0 0 0  194.5   19.6 24.187FF1994   4596 36.9
00899B    199410547    0  0.0     0 0 0   49.3    7.7119.076FF2024   4284 44.8
01083A    199413555    0  0.0     0 0 0  174.9   23.6  3.948FF2024   1466 28.0
01083A    199514554    0  0.0     0 0 0  233.2   25.5  3.644FF2025   1510 28.0
01363A    1994 1755    0  0.0     0 0 0   95.7   15.2  7.422FF2024   1641 34.3
01363A    1995 2754    0  0.0     0 0 0  266.0   24.9  4.528FF2025   1691 34.3
01363A    1996 7744    0  0.0     0 0 0  266.0   23.0  4.889FF2026   1743 34.3
02431     1994 4574    0  0.0     0 0 0 2778.1  291.6  2.174FF2039104333157.5
02456AWBL1994 3456     0  0.0     0 0 0  145.3   26.0  6.677FF2039   3720 40.7
02456AWBL1995 8445     0  0.0     0 0 0  173.0   25.9  6.719FF2040   3833 40.7
02456AWBL1997 9334     0  0.0     0 0 0  361.2   32.4  5.358FF2042   4071 40.7
03451     199411443    0  0.0     0 0 0  140.3   15.8 25.528FF2009   2819 28.0
03451.    1995 9343    0  0.0     0 0 0  145.6   14.9 26.939FF2010   2905 28.0
04540CSBL199415555     0  0.0     0 0 0  350.0   47.0 10.029FF2039  13379 41.3
04540CSBL199612444     0  0.0     0 0 0  324.1   38.8 12.150FF2041  14206 41.3
04858AWBL1996 1655     0  0.0     0 0 0  128.3   15.0 38.981FF2026   7892 38.0
04858AWBL1997 6645     0  0.0     0 0 0  128.3   13.9 42.084FF2027   8133 38.0
06934     1994 5747    0  0.0     0 0 0   25.0   10.2 10.137FF2024   1532 43.5
```

Figure 8.14 Cost.out

Table 8.6 Cost.out Format

Position	Description
1-5	bridge number from the bridge database file
6-9	bridge designation
10-13	year of action
14-15	action code
16	superstructure condition at time of action
17	substructure condition at time of action
18	deck condition at time of action
19-24	exception bridge length passed from DTREE via COST
25-29	exception bridge width
30-35	exception approach length
36-37	exception vertical clearance (ft)
38-39	exception vertical clearance (in)
40-46	cost of project in base year dollars
47-53	EUAC in base year dollars, assuming project specified is done in the action year
54-60	CEF in base year dollars
61	flag set to TRUE if any exception information was used by the COST program
62	flag set to FALSE if action estimated by DTREE o NOTE: program has been changed so that flag will always be false
63-66	year bridge will be replaced based on the action recommended and the life cycle model
67-72	estimated ADT at the year of action specified
73-77	clear deck width after action specified above

```
"COST" "PROGRAM" "REPORT"

"RUNFILE:"  "test.run     " "ANAL YR:" 1992


"BR Number" "Route #" "District" "County" "Act Year" "Act Code" "Action" "Cost" "EUAC" "CEF" "Exe Used" "Year" "Action" "Cost""Year" "Action"
"Cost""Year" "Action" "Cost"
"---------""---------""---------""---------""---------""---------""---------""---------""---------""---------""---------""---------""---------"-
--------""---------""---------""---------""---------""---------""---------""---------"
"00652     " "0007" 5  39 1994 16 "Replace          "   206.2   20.8  25.64 "F" 2014  3   103.6 2029  3   103.6 2044 16   206.2
"00899B    " "0024" 4  56 1994 10 "Sub&Sup Rehab    "    52.3    8.2 126.26 "F" 2009  3    49.9 2024 16   134.2 2044  3    49.9
"01083A    " "0056" 5  78 1994 13 "Widen&Deck Rehab"   185.5   25.0   4.19 "F" 2009  3   125.6 2024 16   309.1 2044  3   125.6
"01083A    " "0056" 5  78 1995 14 "Widen&Deck Repl "   247.3   27.1   3.86 "F" 2010  3   125.6 2025 16   309.1 2045  3   125.6
"01363A    " "0227" 3  89 1994  1 "Deck Rehab       "   101.4   16.1   7.87 "F" 2009  3   106.5 2024 16   297.6 2044  3   106.5
"01363A    " "0227" 3  89 1995  2 "Deck Replace     "   282.1   26.4   4.80 "F" 2010  3   106.5 2025 16   297.6 2045  3   106.5
"01363A    " "0227" 3  89 1996  7 "Sub RH Deck Repl"   282.1   24.4   5.18 "F" 2011  3   106.5 2026 16   297.6 2046  3   106.5
"02431     " "1065" 3  49 1994  4 "Sup RH Deck Repl"  2945.6  309.2   2.30 "F" 2009  2  2805.3 2024  3   577.3 2039 16  2019.4
"02456AWBL" "0024" 2  52 1994  3 "Super&Deck Rehab"   154.1   27.6   7.08 "F" 2009  2   408.1 2024  3   154.1 2039 16   383.0
"02456AWBL" "0024" 2  52 1995  8 "Sub,Sup&Deck RH "   183.4   27.4   7.12 "F" 2010  2   408.1 2025  3   154.1 2040 16   383.0
"02456AWBL" "0024" 2  52 1997  9 "Sub RH Sup Repl "   383.0   34.4   5.68 "F" 2012  2   408.1 2027  3   154.1 2042 16   383.0
"03451     " "0331" 4  50 1994 11 "Sub&Sup RH,Dk RP"   148.7   16.7  27.07 "F" 2009 16   154.4 2029  3    77.0 2044  3    77.0
"03451     " "0331" 4  50 1995  9 "Sub RH Sup Repl "   154.4   15.8  28.56 "F" 2010 16   154.4 2030  3    77.0 2045  3    77.0
"04540CSBL" "1069" 2   2 1994 15 "Raise/Lower Pvmt"   371.1   49.9  10.63 "F" 2009  2   598.3 2024  3   225.9 2039 16   482.5
"04540CSBL" "1069" 2   2 1996 12 "Super Replace    "   343.6   41.2  12.88 "F" 2011  2   598.3 2026  3   225.9 2041 16   482.5
"04858AWBL" "0024" 2  52 1996  1 "Deck Rehab       "   136.1   15.9  41.33 "F" 2011  3   142.9 2026 16   250.8 2046  3   142.9
"04858AWBL" "0024" 2  52 1997  6 "Sub&Deck Rehab   "   136.1   14.8  44.62 "F" 2012  3   142.9 2027 16   250.8 2047  3   142.9
"06934     " "0252" 5  24 1994  5 "Substruct Rehab "    26.5   10.8  10.75 "F" 2009  3   106.1 2024 16   280.8 2044  3   106.1
"=========="

" Summary  "
"Records Processed:"     18
"Total Errors    : "      0

"Total Cost ($1000)"     6339.3
"Total EUAC ($1000)"      711.7
"Total CEF  ($1000)"      375.8

"AVE   Cost ($1000)"      352.2
"AVE   EUAC ($1000)"       39.5
"AVE   CEF  ($1000)"       20.9
```

Figure 8.15 Cost.prn

by DTREE and calculates the life cycle costs (EUAC) and cost effectiveness factor (CEF) for each bridge. The "look ahead" text report provides repair costs, in analysis year dollars, for projects anticipated in the future. If no action is recommended by DTREE (bridges 04858AWBL and 04216BSBL, for example), a life cycle cost analysis will not be performed. In addition, these records will not be used by subsequent programs.

8.5 RANK

The procedure for running RANK is shown in Figure 8.16. The corresponding output files are provided in Figures 8.17 through 8.20. See Tables 8.7, 8.8 for the format of RANK output files. At this point, projects can be selected on the basis of their disutility; as the disutility increases, the urgency for repair increases.

The disutilities are computed individually for each utility function, and then weighted and added together to yield the overall disutility. The "utility" report (Figure 8.17) lists the individual disutilities before a repair is made.

8.6 OPT

The procedure for running OPT is shown in Figure 8.21. The corresponding output files are provided in Figures 8.22 through 8.24. By running OPT, projects are optimized one year at a time over a four-year period. The report (Figures 8.22-8.24)

```
[A:\BRIDGE\PROG]rank

PROJECT RANKING PROGRAM
=======================

Enter runfile name [default] a:\bridge\run\runtest.run

Enter year of analysis: 1992

Enter report style (n)ormal (u)tility [u] u

SelectOneAction...:    18 actions.    10 selected and     8 rejected.   GetADTStats: 0
errs,  10 ADTEs used to compute mean=27.020     stdev=37.299

Quitting rank with successful completion.
****done****
Press any key to continue...

[A:\BRIDGE\PROG]
```

Figure 8.16 Running RANK

```
RUNFILE: test.run                   RANK          PROGRAM        REPORT                  YEAR OF ANALYSIS: 1992

 Bridge   Route  D  Co.   Act   Act                    Cost    EUAC   E Util Util Olta Cost  Rem.  Strc  Clr. Vert  Inv   Det.
num   des   #    s  Code  Year  Code     Action       ($1000) $1000   x Now  Aftr Util Eff.  Life  Cond  Deck Clr.  Rtng  Len.
------------------------------------------------------------------------------------------------------------------------------
00652      0007  5   39   1994   16 Replace            206.2   20.8 F  57    7   50   51   100   100    9    0    10     0
00899B     0024  4   56   1994   10 Sub&Sup Rehab       52.3    8.2 F  17    9    8    6     0    80    0    0     0     0
01083A     0056  5   78   1994   13 Widen&Deck Rehab   185.4   25.0 F  26   18    8   60     0    60   83    0     0     0
01363A     0227  3   89   1994    1 Deck Rehab         101.5   16.1 F  21   13    8   59     0    60   31    0     0     0
02431      1065  3   49   1994    4 Sup RH Deck Repl  2945.6  309.2 F  22   10   12   61     0    80    0    0     0     0
02456AWBL  0024  2   52   1995    8 Sub,Sup&Deck RH    183.4   27.5 F  22    6   16   59     0    80    0    0     0     0
03451      0331  4   50   1995    9 Sub RH Sup Repl    154.4   15.8 F  32   12   20   49     0   100   83    0     0     0
04540CSBL  1069  2    2   1996   12 Super Replace      343.6   41.1 F  36   28    8   56     0    80   73   83     0     0
04858AWBL  0024  2   52   1997    6 Sub&Deck Rehab     136.0   14.7 F  23   15    8   42     0    80    0    0    20     0
06934      0252  5   24   1994    5 Substruct Rehab     26.5   10.8 F  22   14    8   57     0    80    0    0     0     0
------------------------------------------------------------------------------------------------------------------------------
SelectOneAction...:    18 actions.    10 selected and     8 rejected.
GetADTStats: 0 errs,  10 ADTEs used to compute mean=27.020     stdev=37.299
```

Figure 8.17 Rank.txt

```
RUNFILE: runfile.run    RANK     PROGRAM     LOG        YEAR OF ANALYSIS: 1992
------------------------------------------------------------------------------
 SelectOneAction...:    18 actions.    10 selected and     8 rejected.
 GetADTStats: 0 errs,  10 ADTEs used to compute mean=27.020    stdev=37.299
------------------------------------------------------------------------------
```

Figure 8.18 Rank.log

Table 8.7 Description of Rank.txt Utility Report

Heading	Description
Bridge num and des	Bridge number and designation
Route #	Route number
Ds	District code
Co. Code	County code
Act Year	Action year
Act Code	Action code
Action	Action
Cost ($1000)	Base year cost converted to analysis year dollars, in thousands
EUAC $1000	Equivalent uniform annual cost, in analysis year dollars
Ex? (T or F)	T: Exception record used F: Exception record not used
Util Now	Total disutility of bridge before repair, for action year
Util Aftr	Total disutility after repair, for action year
Dlta Util	Change in total disutility before and after repair, for action year
Cost Eff.[1]	Cost effectiveness disutility function value
Rem. Life[1]	Remaining life disutility function value
Strc Cond[1]	Structural condition disutility function value
Clr. Deck[1]	Clear deck width disutility function value
Vert Clr.[1]	Vertical clearance disutility function value
Inv Rtng[1]	Inventory rating disutility function value
Det. Len.[1]	Detour length disutility function value

[1] These disutility values represent the disutility before a repair is made.

```
1   5    10    15    20    25    30    35    40    45    50    55    60    65    70    75    80    85    90    95   100   105   110   115   120
----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+
00652      199416345      0  0.0      0 0 0  206.2    19.6 24.187FF571994   4596 75057575757 7 7 7 7  4596  4735  4880  5028
00899B     199410547      0  0.0      0 0 0   52.3     7.7119.076FF172024   4284 9 817172121 9 91313  4284  4415  4549  4688
01083A     199413555      0  0.0      0 0 0  185.4    23.6  3.948FF262024   146618 8263030301822222  1466  1510  1556  1604
01363A     1994 1755      0  0.0      0 0 0  101.5    15.2  7.422FF212024   164113 82125252513171717  1641  1691  1743  1796
02431      1994 4574      0  0.0      0 0 0 2945.6   291.6  2.174FF22203910433310122222222210101010104333107510110784114158
02456AWBL1995 8445        0  0.0      0 0 0  183.4    25.9  6.719FF222040   3833 61622222226 6 6 610  3720  3833  3950  4071
03451      1995 9343      0  0.0      0 0 0  154.4    14.9 26.939FF322010   2905122032323353812121212  2819  2905  2993  3084
04540CSBL199612444        0  0.0      0 0 0  343.6    38.8 12.150FF362041 1420628 83232363624242828 13379 13787 14206 14639
04858AWBL1997 6645        0  0.0      0 0 0  136.0    13.9 42.084FF232027   813315 815191923 7111115  7433  7659  7892  8133
06934      1994 5747      0  0.0      0 0 0   26.5    10.2 10.137FF222024   153214 82226262614181818  1532  1578  1627  1676
```

Figure 8.19 Rank.out

```
"RANK" "PROGRAM" "REPORT"

"RUNFILE:" "test.run     " "ANALYSIS YEAR:" 1992

         "Bridge #" "Route #" "District" "County" "Act Year" "Act Code"   "Action" "Cost" "EUAC" "Exe Used" "Util Now" "Util Aftr"         "Olta
Utl" "Cost Eff" "Rem Life" "Strc Cond" "Clr. Deck"          "Vert Clr" "Inv Rating" "Det Len"
"---------""---------""---------""---------""---------""---------""---------""---------""---------""---------""---------""---------""---------"
---------""---------""---------""---------""---------""---------""---------"
"00652     " "0007"  5  39 1994 16 "Replace          "   206.2   20.8 "F"  57   7  50  51 100 100   9   0  10   0
"00899B    " "0024"  4  56 1994 10 "Sub&Sup Rehab    "    52.3    8.2 "F"  17   9   8   6   0  80   0   0   0   0
"01083A    " "0056"  5  78 1994 13 "Widen&Deck Rehab"   185.4   25.0 "F"  26  18   8  60   0  60  83   0   0   0
"01363A    " "0227"  3  89 1994  1 "Deck Rehab       "   101.5   16.1 "F"  21  13   8  59   0  60  31   0   0   0
"02431     " "1065"  3  49 1994  4 "Sup RH Deck Repl"  2945.6  309.2 "F"  22  10  12  61   0  80   0   0   0   0
"02456AWBL" "0024"   2  52 1995  8 "Sub,Sup&Deck RH "   183.4   27.5 "F"  22   6  16  59   0  80   0   0   0   0
"03451     " "033'"  4  50 1995  9 "Sub RH Sup Repl "   154.4   15.8 "F"  32  12  20  49   0 100  83   0   0   0
"04540CSBL" "1069"   2   2 1996 12 "Super Replace    "   343.6   41.1 "F"  36  28   8  56   0  80  73  83   0   0
"04858AWBL" "0024"   2  52 1997  6 "Sub&Deck Rehab  "   136.0   14.7 "F"  23  15   8  42   0  80   0   0  20   0
"06934     " "0252"  5  24 1994  5 "Substruct Rehab "    26.5   10.8 "F"  22  14   8  57   0  80   0   0   0   0
"---------" "---------" "---------" "---------" "---------" "---------" "---------" "---------" "---------" "---------" "---------" "---------"
"---------" "---------" "---------" "---------" "---------" "---------" "---------" "---------" "---------"
"SelectOneAction...:    18 actions.    10 selected and     8 rejected.          "
"GetADTStats: 0 errs,  10 ADTEs used to compute mean=27.020     stdev=37.299     "
```

Figure 8.20 Rank.prn

Table 8.8 Rank.out Format

Position	Description
1-5	bridge number in the bridge database
6-9	bridge designation
10-13	year of action
14-15	action code
16	superstructure condition at the year of action
17	substructure condition at the year of action
18	deck condition at the year of action
19-24	exception bridge length passed from DTREE via COST
25-29	exception clear deck width
30-35	exception approach length
36-37	exception vertical clearance (ft)
38-39	exception vertical clearance (in)
40-46	repair cost in analysis year dollars from COST
47-53	base EUAC from COST
54-60	base CEF from COST
61	exception used flag from COST
62	if TRUE, repair was derived from the life cycle instead of DTREE. RANK will not process any records set to TRUE, so all output in this field will be FALSE
63-64	disutility of project if done in year of action
65-68	year of replacement from COST
69-74	estimated ADT in the year of action
75-76	estimated disutility after the project is done (if done in the year of action)
77-78	change in disutility after project completion
79-80	disutility of project 2 years after analysis year
81-82	disutility of project 3 years after analysis year
83-84	disutility of project 4 years after analysis year
85-86	disutility of project 5 years after analysis year
87-88	estimated disutility after project completion if done 2 years after year of analysis
89-90	estimated disutility after project completion if done 3 years after year of analysis

Table 8.8, continued

Position	Description
91-92	estimated disutility after project completion if done 4 years after year of analysis
93-94	estimated disutility after project completion if done 5 years after year of analysis
95-100	estimated ADT 2 years after year of analysis
101-106	estimated ADT 3 years after year of analysis
107-112	estimated ADT 4 years after year of analysis
113-118	estimated ADT 5 years after year of analysis

```
[A:\BRIDGE\PROG]opt
...

Input analysis year (same as DTREE): 1992
   ...

OPTIMIZATION PROGRAM
====================
Enter year of optimization
1994
Enter the estimated budget for each year ($1000)
 YEAR          1994:  1000
    ...

 OPTIMIZATION PROGRAM
====================
Enter year of optimization
1995
Enter the estimated budget for each year ($1000)
 YEAR          1995:  1000
    ...

OPTIMIZATION PROGRAM
====================
Enter year of optimization
1996
Enter the estimated budget for each year ($1000)
 YEAR          1996:  1000
    ...

 OPTIMIZATION PROGRAM
====================
Enter year of optimization
1997
Enter the estimated budget for each year ($1000)
 YEAR          1997:  1000
    ...
```

Figure 8.21 Running OPT

provides a year by year listing of the projects selected by OPT. The year indicated in the output may not correspond to the year of optimization, signifying that the funding for a previous year was not adequate and the project was carried over.

The o/p of the optimization program is a year by year listing of the selected bridges in COST o/p format, given by CHOSEN_1.OUT, CHOSEN_2.OUT, CHOSEN_3.OUT, and CHOSEN_4.OUT. They are located in the \bridge\reports directory, and can be viewed as described in Chapter 3. These files are listed in Figures 8.22 through 8.24. CHOSEN_4.OUT is not present since all bridges had been selected before the fourth year.

```
00652     199416345    0  0.0     0 0 0  194.5    19.6 24.187FF1994  4596 36.9
008998    199410547    0  0.0     0 0 0   49.3     7.7119.076FF2024  4284 44.8
01083A    199413555    0  0.0     0 0 0  174.9    23.6  3.948FF2024  1466 28.0
01363A    1994 1755    0  0.0     0 0 0   95.7    15.2  7.422FF2024  1641 34.3
02456AWBL1994 3456     0  0.0     0 0 0  145.3    26.0  6.677FF2039  3720 40.7
03451     199411443    0  0.0     0 0 0  140.3    15.8 25.528FF2009  2819 28.0
```

Figure 8.22   chosen_1.out

```
04540CSBL199415555     0  0.0     0 0 0  350.0    47.0 10.029FF2039 13379 41.3
```

Figure 8.23   chosen_2.out

```
04858AWBL1996 1655     0  0.0     0 0 0  128.3    15.0 38.981FF2026  7892 38.0
```

Figure 8.24   chosen_3.out

LIST OF REFERENCES

Federal Highway Administration [FHWA] (1987), "Bridge Management Systems," U.S. Department of Transportation, Demonstration Project No. 71, Report No. FHWA-DP-71-01.

Federal Highway Administration [FHWA] (1992), "Price Trends for Federal-Aid Highway Construction, Third Quarter, 1991," U.S. Department of Transportation, Publication No. FHWA-PD-92-009.

Federal Highway Administration [FHWA] (1988), "Recording and Coding Guide for the Structure Inventory and Appraisal of the Nation's Bridges," U.S. Department of Transportation, Report No. FHWA-ED-89-044.

International Business Machines [IBM] (1987), "IBM FORTRAN/2 Compile, Link and Run."

International Business Machines [IBM] (1991), "IBM Operating System/2 Standard Edition Version 1.3 Getting Started."

Jiang, Yi and Sinha, Kumares C. (1989), "The Development of Optimal Strategies for Maintenance, Rehabilitation and Replacement of Highway Bridges, Final Report Vol. 6: Performance Analysis and Optimization," Report No. FHWA/IN/JHRP-89/13, Joint Highway Research Project, School of Civil Engineering, West Lafayette, Indiana.

Saito, Mitsuru and Sinha, Kumares C. (1989a), "The Development of Optimal Strategies for Maintenance, Rehabilitation and Replacement of Highway Bridges, Final Report Vol. 4: Cost Analysis," Report No. FHWA/IN/JHRP-89/11, Joint Highway Research Project, School of Civil Engineering, West Lafayette, Indiana.

Saito, Mitsuru, and Sinha, Kumares C. (1989b), "The Development of Optimal Strategies for Maintenance, Rehabilitation and Replacement of Highway Bridges, Final Report Vol. 5: Priority Ranking Method," Report No. FHWA/IN/JHRP-89/12, Joint Highway Research Project, School of Civil Engineering, West Lafayette, Indiana.

Appendix A DTREE Flow Charts

* Refer to separate flow chart.

(1) Convert Recording and Coding Guide codes to codes used by the program. For example, assign default values for approach lengths. See Chapter 2 for more information.

(2) A record was located in the exception file that matches the bridge number and designation of a record in the bridge database record.

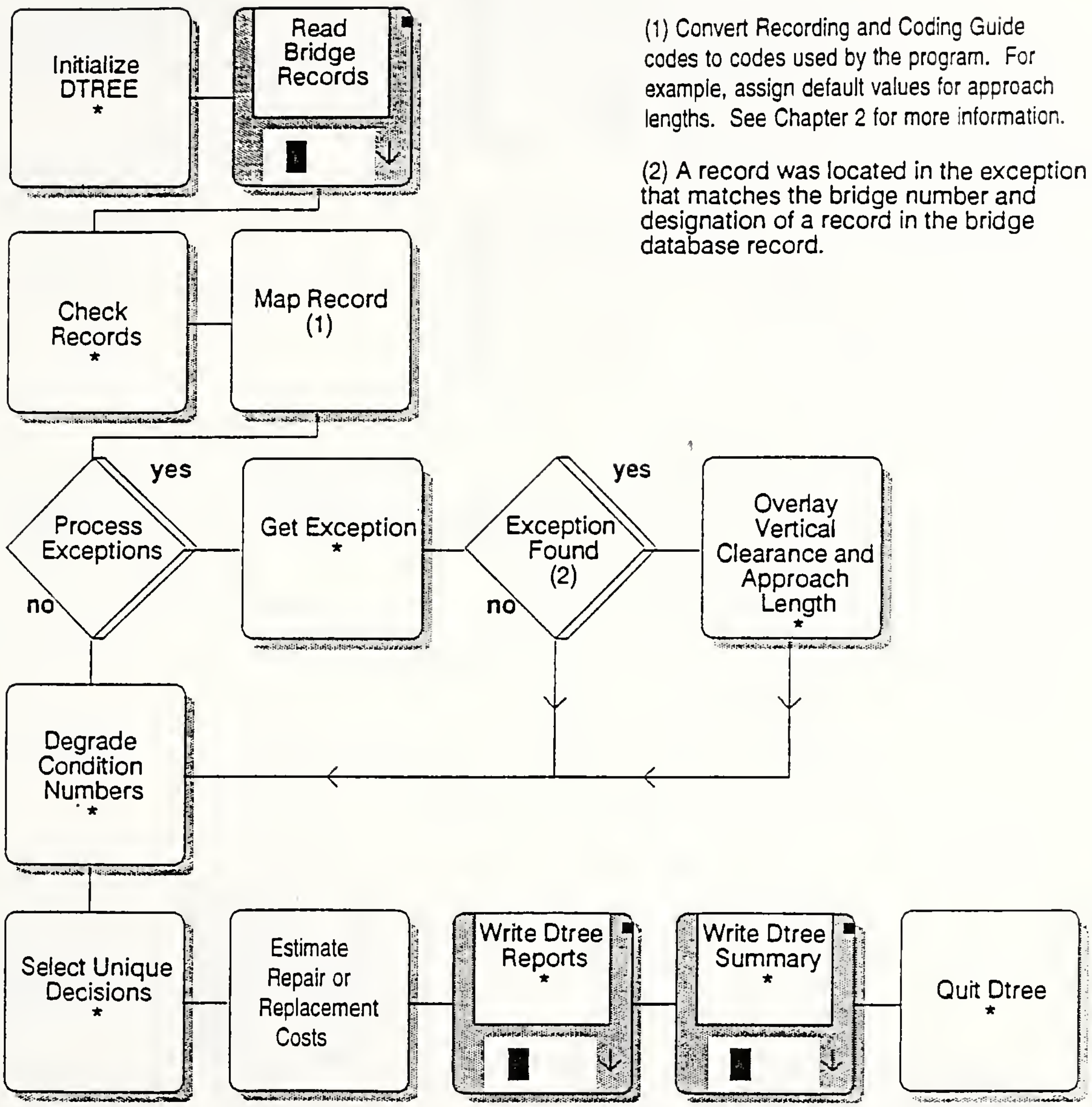
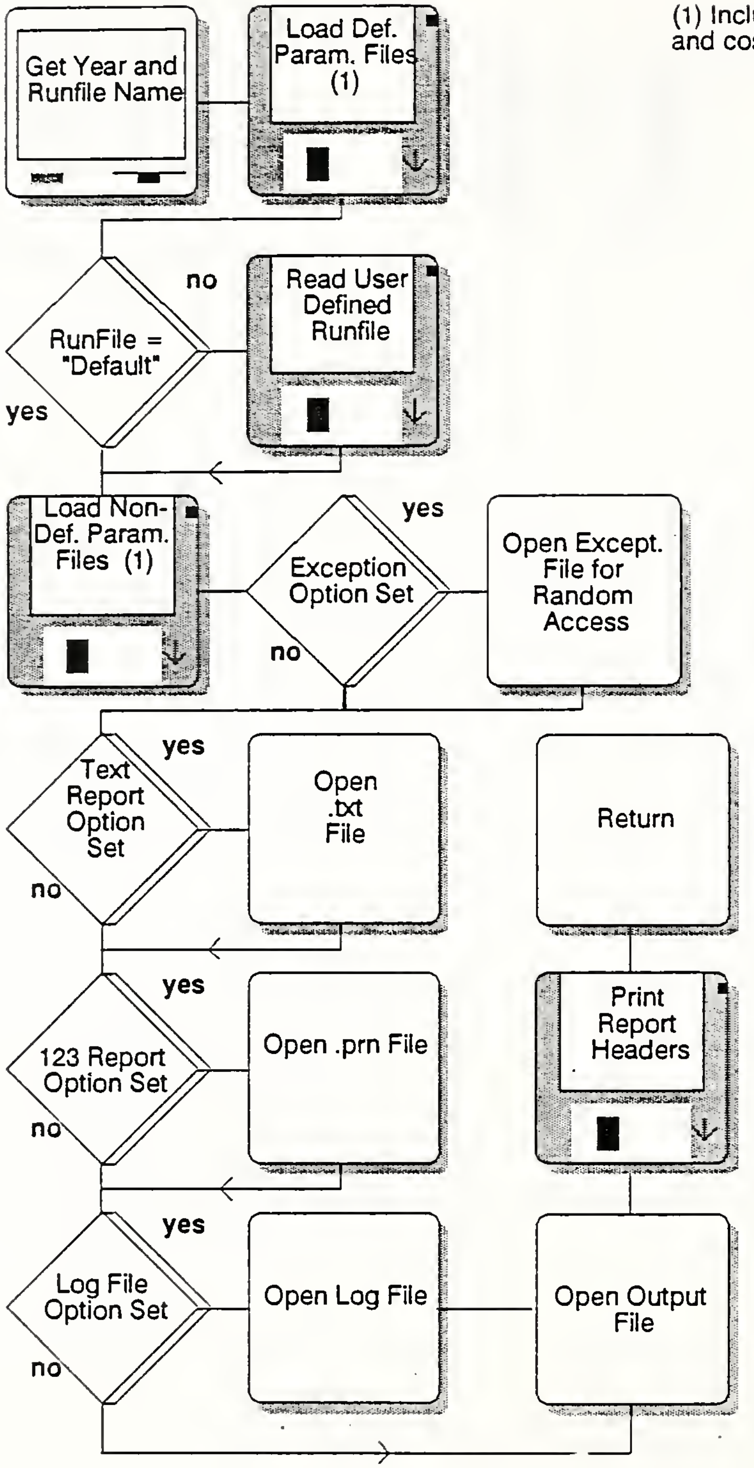
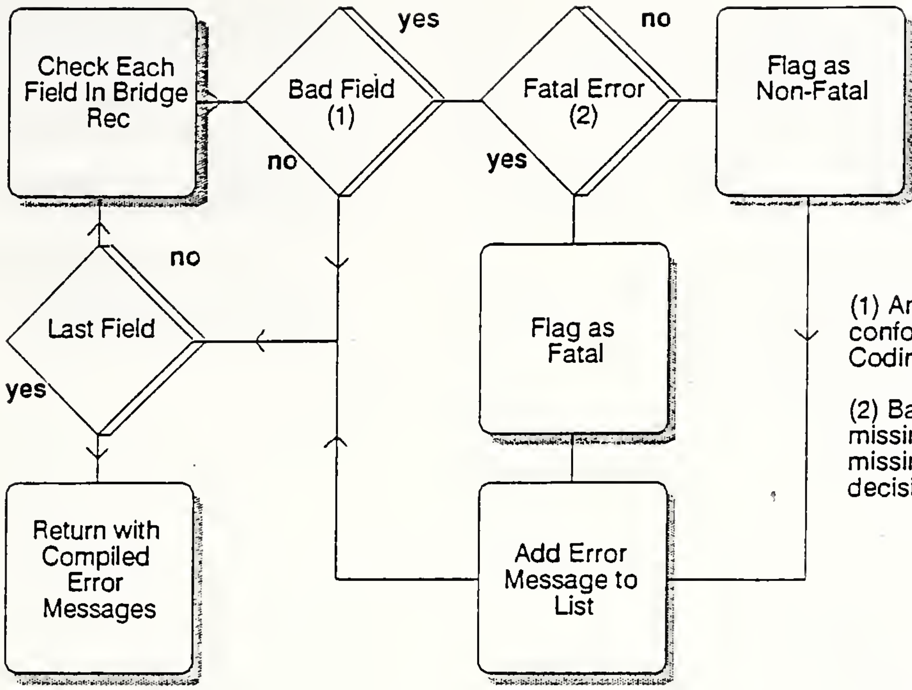


Figure A.1 DTREE: Main Program



(1) Includes dollar conversion, decision tree, and cost files.

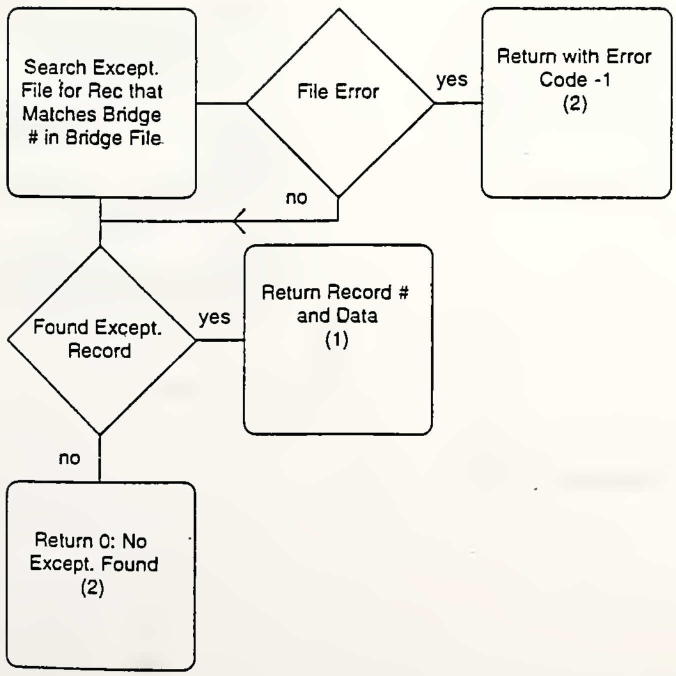
Figure A.2 Initialize DTREE



(1) Any field with data not conforming to the Recording and Coding Guide.

(2) Bad condition numbers, missing lengths or widths, or other missing or bad data which prevent decision tree from functioning.

Figure A.3 Check Record

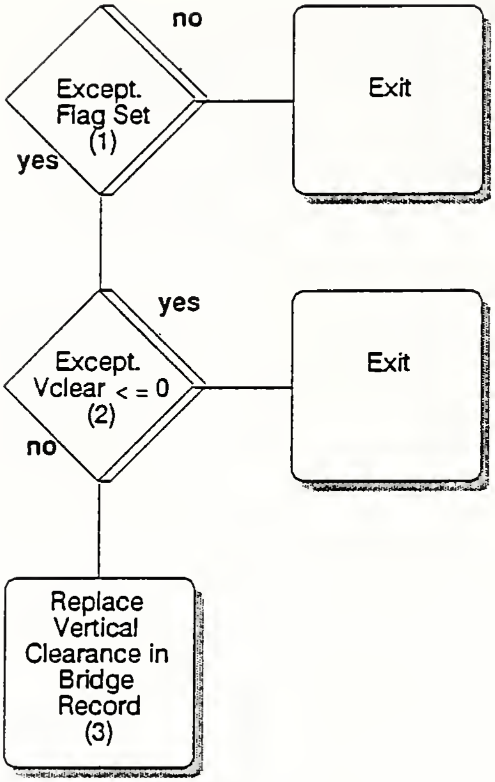


(1) Includes exception action, year, length, width, approach length (feet), and vertical clearance (feet and inches).

(2) If code is < =0, then the main program will not attempt to process exceptions for this bridge.

See Chapters 6 and 7 for more information about exception files.

Figure A.4 Get Exception

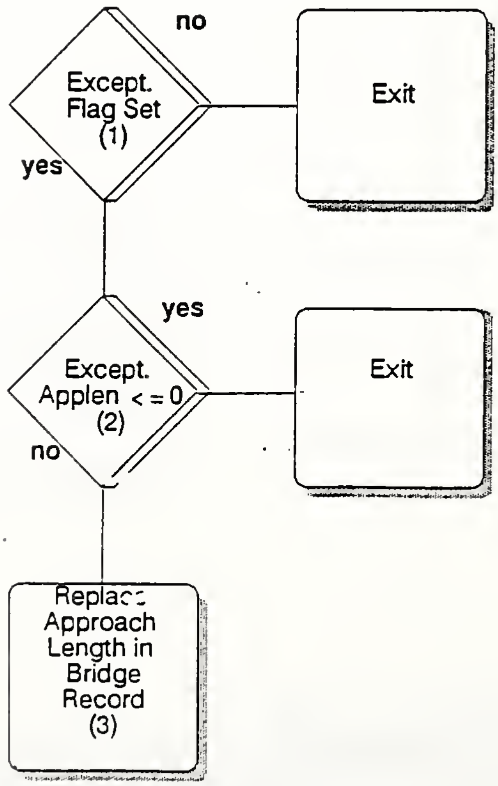


(1) The "process exceptions" option has been set to yes and an exception has been found for the current bridge record.

(2) Vertical clearance units: feet and inches.
- If Vclear < = 0, a default value is set by th program.
- If Vclear > 0, that value in the field is used as the vertical clearance.

(3) For more information, see DTREE: Main Program, "Map Record" or program listings bdata.for and map.for.

Figure A.5 Overlay Vertical Clearance

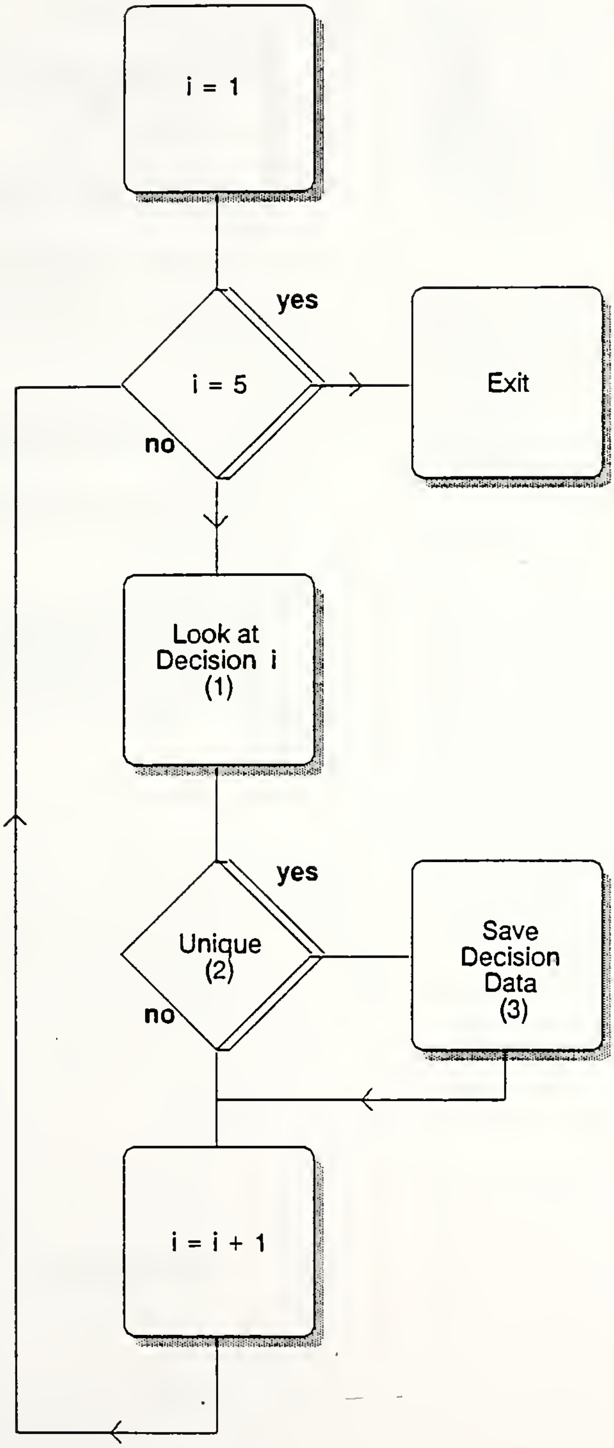


(1) The "process exceptions" option has been to yes and an exception has been found for th current bridge record.

(2) Approach length units: feet.
- If Applen < = 0, a default value is set by the program.
- If Applen > 0, that value in the field is used the approach length.

(3) For more information, see DTREE: Main Program, "Map Record" or program listings bdata.for and map.for.

Figure A.6 Overlay Approach Length

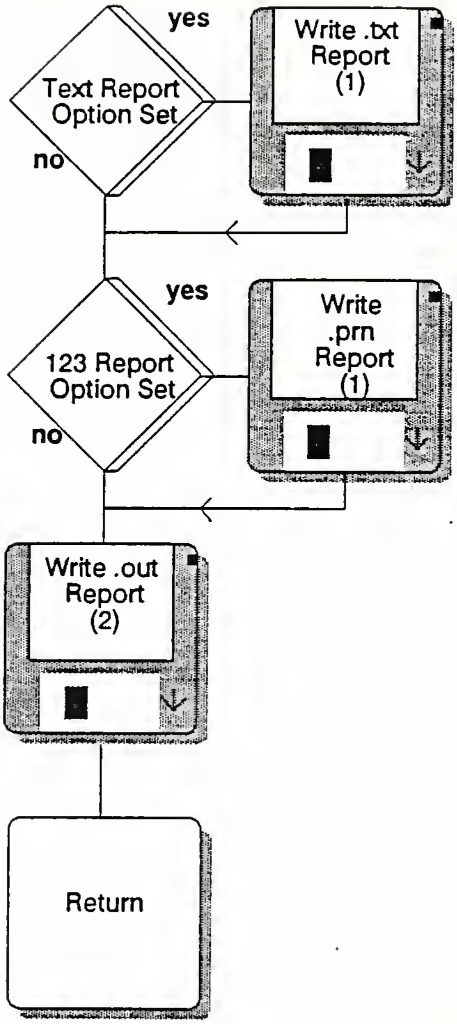


(1) Decisions for a bridge are based on projected condition numbers 2, 3, 4, and 5 years from the year of analysis.

(2) The same decision may have been made for one or all sets of projected condition numbers; save a decision code only when it occurs for the first time.

(3) The year, projected condition numbers, and repair code of the action are saved. If the decision type is "nothing" (no action recommended), the year is set 2 years from the analysis year (now + 2).

Figure A.7 Select Decisions



(1) Reports include:

Standard:

- Bridge number and Designation
- Route number
- District and County
- Repair year
- Repair code and description
- Repair cost, in $1000's, analysis year dollars
- Recommended action from Item 75 of the Recording and Coding Guide
- Exceptions used (True or False)

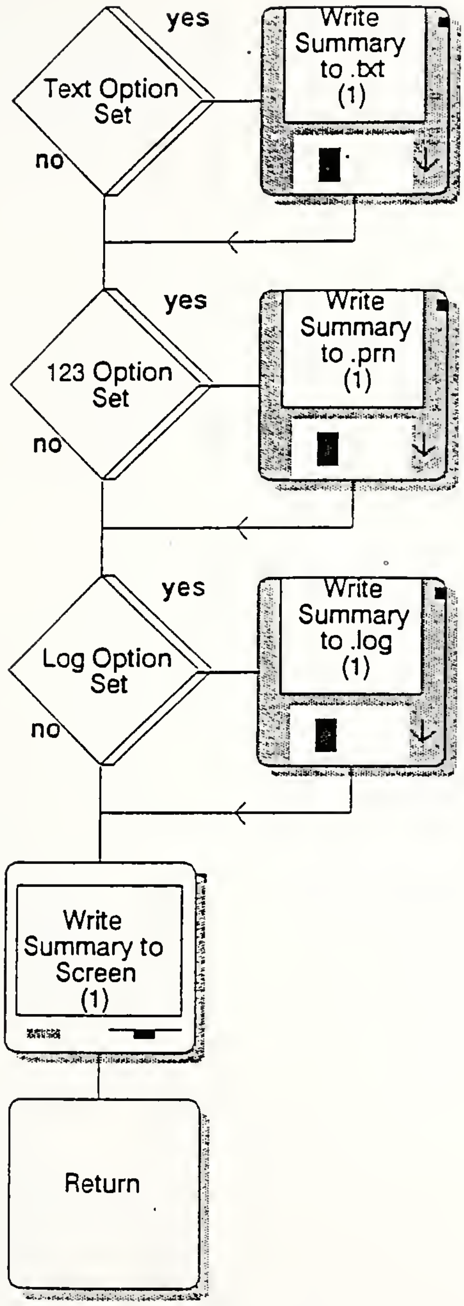
Optional:

- Condition numbers at time of repair
- Exception file information, if any

(2) Output file contains only unique information needed by subsequent programs in the system:

- Bridge number and designation
- Repair year and code
- Condition numbers at time of action
- Exception length, width, approach length, and vertical clearance, if applicable

Figure A.8 Write DTREE Reports



(1) Summary information:

- Number of records successfully processed
- Number of records rejected due to errors
- Total number of each decision selected.

Note that the number of projects selected does not necessarily equal the number of records processed plus the number rejected, since more than one decision can be recommended for each bridge.

Figure A.9 Write DTREE Summary

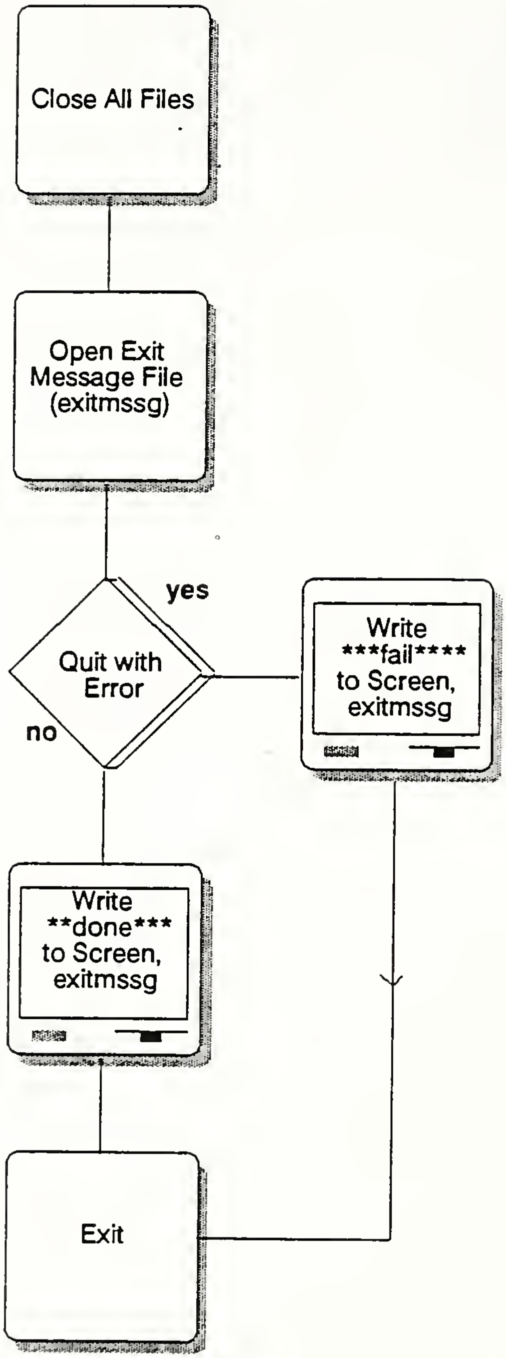
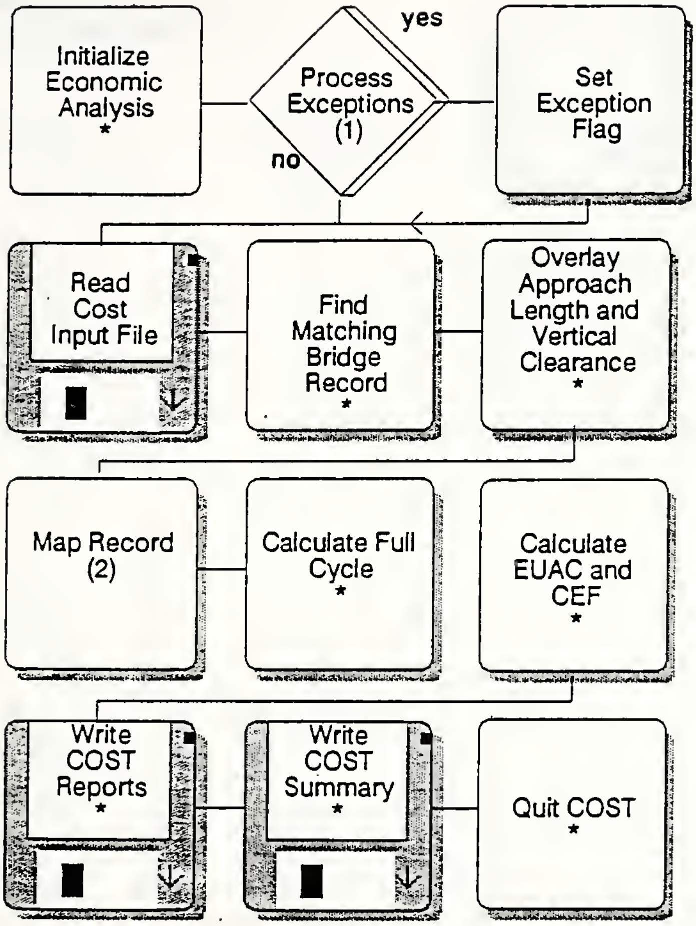


Figure A.10 Quit DTREE

Appendix B COST Flow Charts



* Refer to separate flow chart.

(1) This option is set in the runfile.

(2) Convert Recording and Coding Guide codes to codes used by the program. For example, assign default values for approach lengths. See Chapter 2 for more information.

Figure B.1 COST: Main Program

(1) Includes dollar conversion, cost, and li
cycle model parameter files.

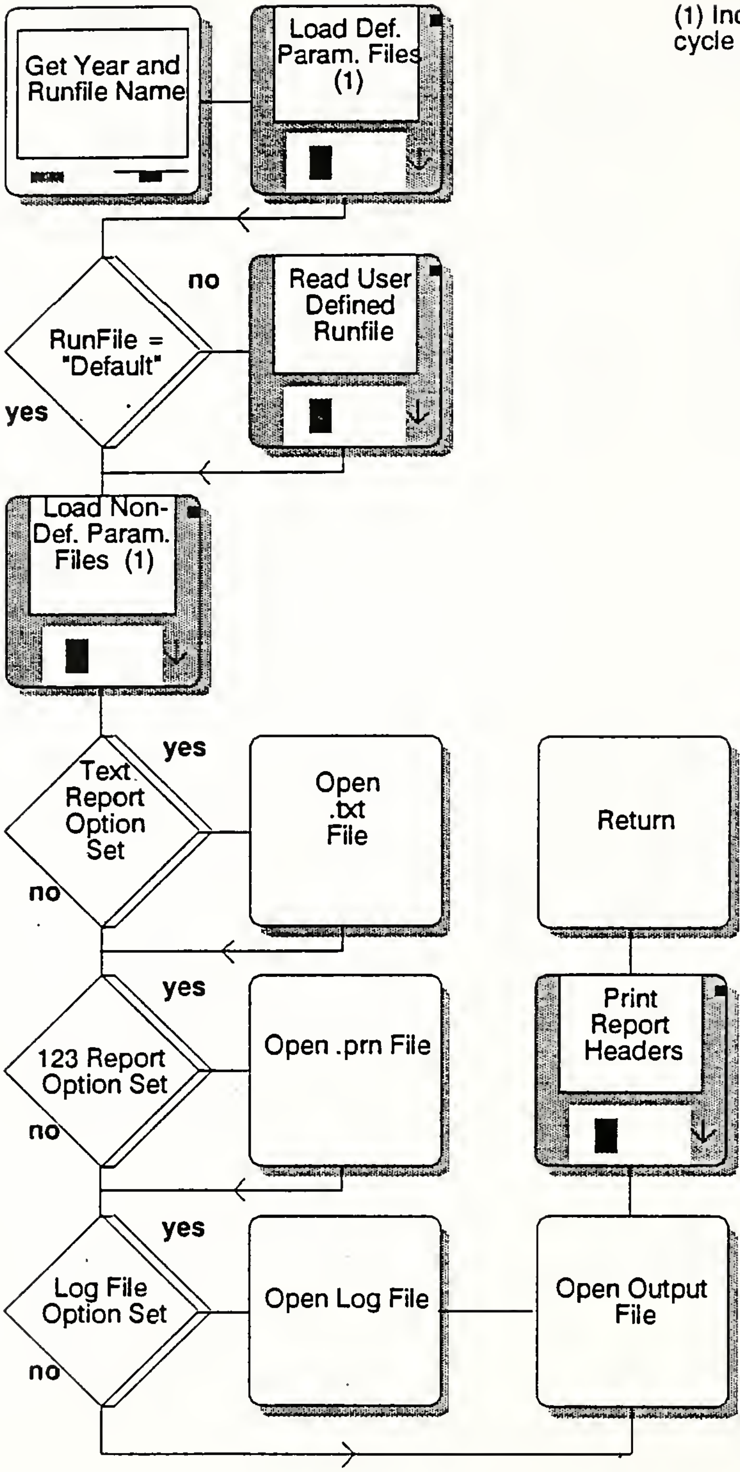
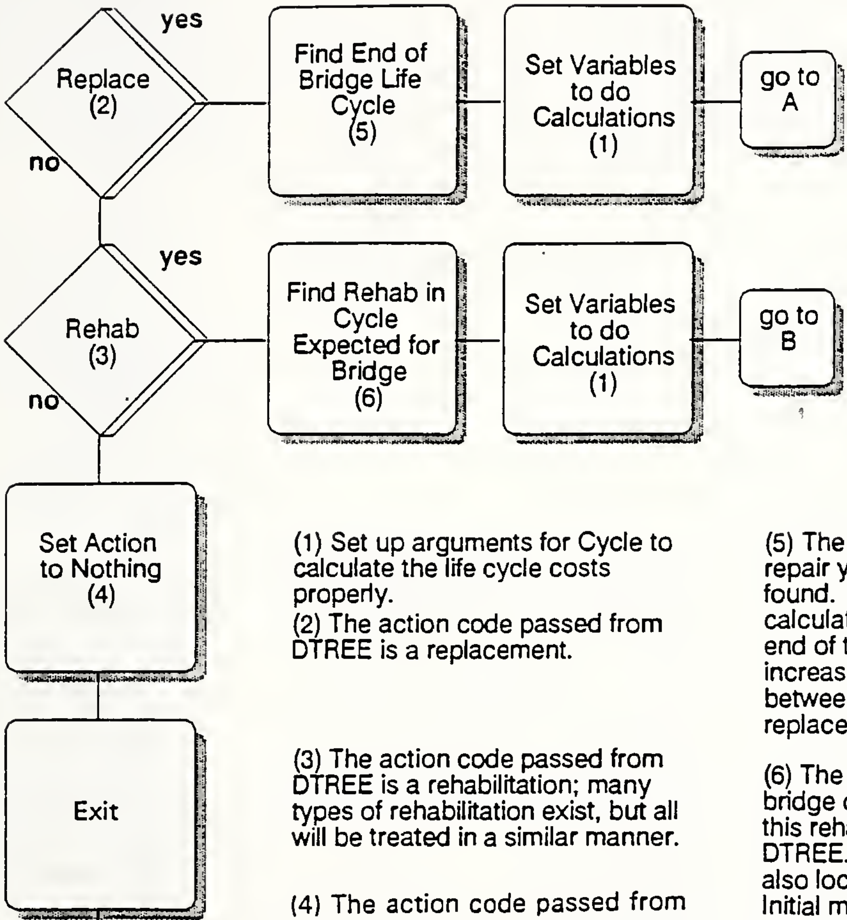


Figure B.2 Initialize COST



(1) Set up arguments for Cycle to calculate the life cycle costs properly.

(2) The action code passed from DTREE is a replacement.

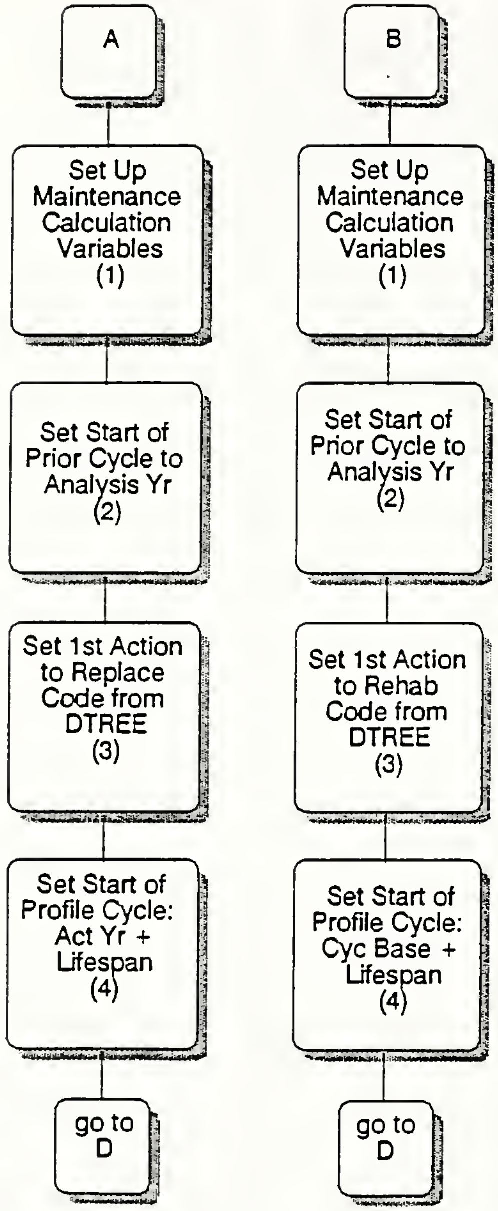
(3) The action code passed from DTREE is a rehabilitation; many types of rehabilitation exist, but all will be treated in a similar manner.

(4) The action code passed from DTREE is either 0 (do nothing) or not recognized as a replacement or rehabilitation. A code of 0 means that DTREE did not recommend a repair, indicating that the bridge is in good condition. Therefore, a life cycle cost analysis will not be performed.

(5) The end of the life cycle and the last repair year before replacement are found. Initial maintenance costs are calculated from the analysis year to the end of the cycle, with any yearly increases based on the time interval between the last repair before replacement and the analysis year.

(6) The rehabilitation expected for a bridge of this age is found, assuming that this rehabilitation is suggested by DTREE. The most recent rehabilitation is also located in the life cycle model. Initial maintenance costs are computed from the analysis year to the time of the repair, based on the time interval between the most recent rehabilitation and the analysis year. If the rehabilitation suggested by DTREE is the first one in the cycle, then the starting point of the life cycle is used as the reference base for maintenance cost calculations.

Figure B.3 Calculate Full Cycle



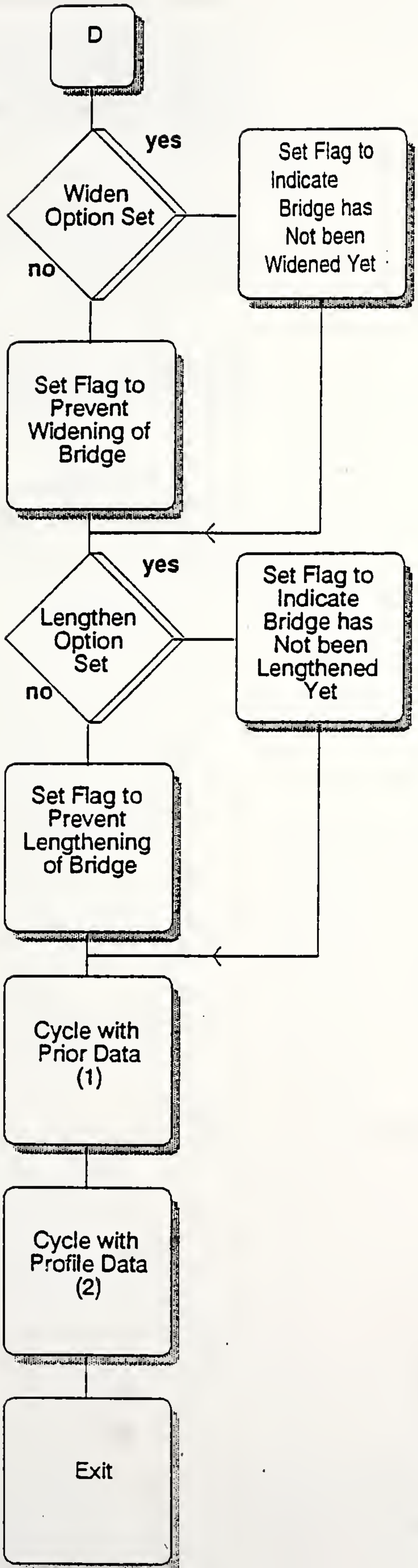
(1) Maintenance costs start at a fixed level after a repair is made. Eventually, these costs may begin to increase. To calculate the maintenance costs between the analysis year (now) and the next repair year, the most recent repair year must be set so that any maintenance cost increases can be accounted for.

(2) The prior cycle includes maintenance costs from "now" until the first action, continuing through the life cycle up to but not including the next replacement. Maintenance costs are calculated for the interval between repairs, which occur at times specified by the life cycle model. For more information, see the Life Cycle Parameter File or bdata.for.

(3) The code (type) of the first action is taken from the ouput of DTREE when non-zero. The life cycle model will typically consist of general repairs, such as bridge replacement, deck rehabilitation, or deck replace. DTREE may specify any of 17 actions, so the code from DTREE is used for the first repair, and those from the life cycle model for later repairs.

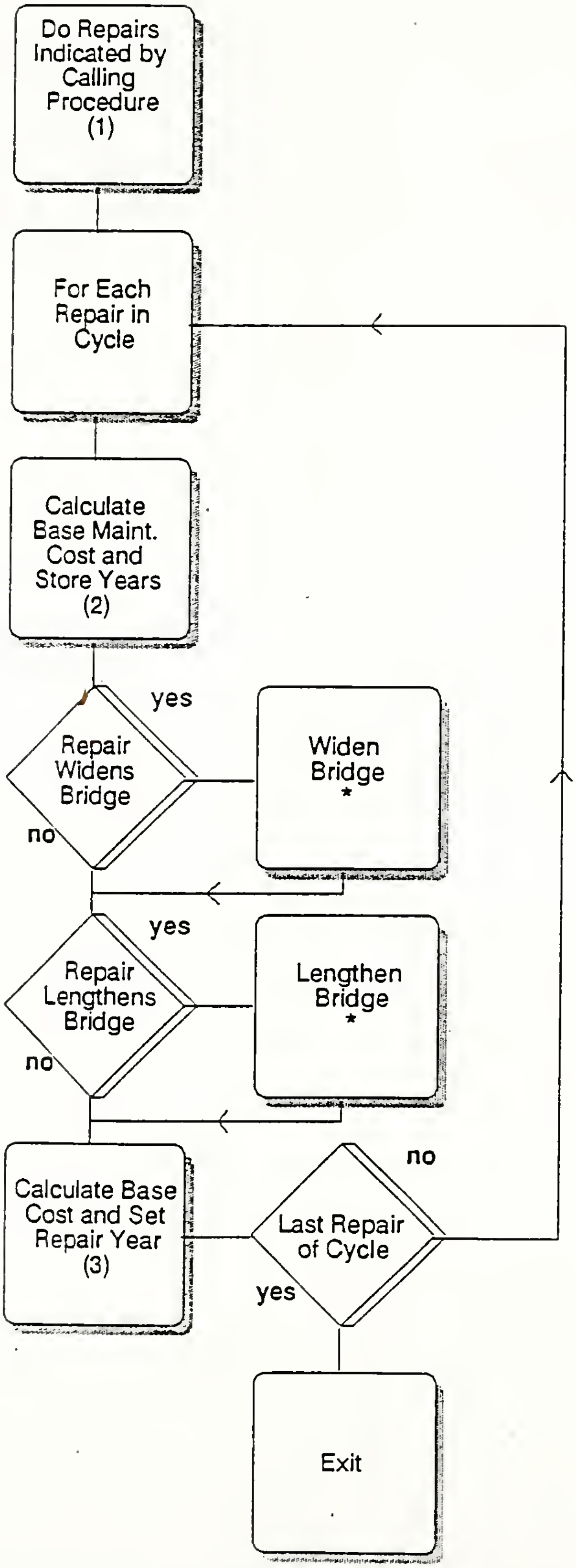
(4) The profile cycle begins with the first replacement in the model after the initial action. If the initial action is a replace, then the profile cycle will begin one full bridge lifespan after the replacement occurs.

Figure B.3, continued



(1) Use variables set by Full Cycle to call Cycle. This step determines repair costs and years, and data needed for maintenance cost computations for the prior cycle.

(2) Use variables set by Full Cycle to call Cycle. This step determines repair costs and years, and data needed for maintenance cost computations for the profile cycle



* Refer to separate flow chart.

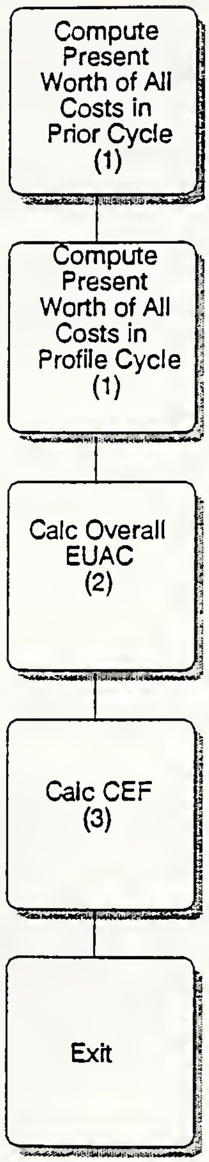
(1) The calling routine (Full Cycle) sets variables passed to Cycle so that it begins at a certain point in the life cycle model and computes costs for all maintenance and repair activities from that point to the end of the model.

(2) The yearly maintenance cost, in base year dollars at the time of the repair, is computed and stored. These costs start out at a fixed level and may increase in subsequent years.

(3) Only the base cost of the repair is stored. The year is also saved so that the present worth of repairs and other economic statistics can be calculated.

See FHWA/IN/JHRP-89/11, Volume 4 and Figure 5.3 for more information about life cycle models.

Figure B.4 Cycle



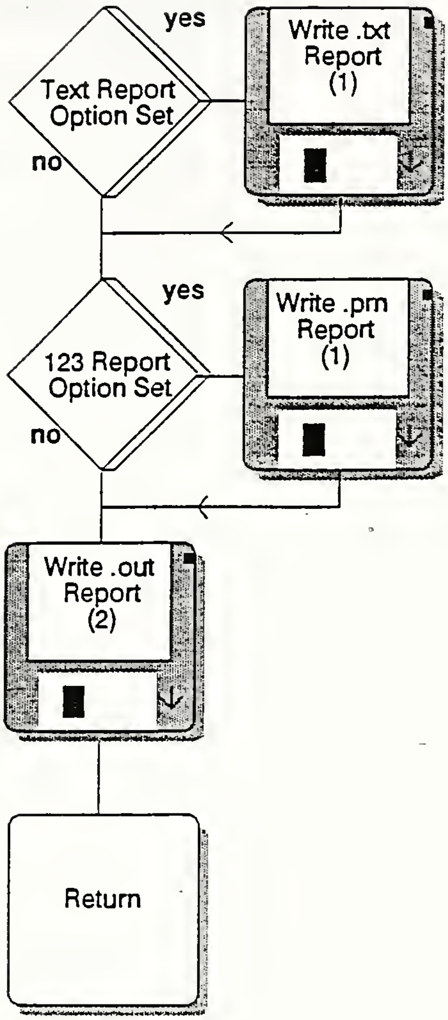
(1) These costs are brought back to the analysis year.

(2) EUAC = equivalent uniform annual cost, in $1000's

(3) CEF = cost effectiveness factor, in veh/$1000/ft^2

See Figure B.3, notes 2 and 4, for more information about the prior and profile cycles.

Figure B.5 Calculate EUAC and CEF



(1) Reports include:

Standard:
- Bridge number and Designation
- Route number
- District and County
- Repair year
- Repair code and description
- Repair cost, in $1000's, analysis year dollars
- EUAC, in $1000's
- CEF, in veh/$1000/ft^2
- Exceptions used (True or False)
- Action produced by life cycle model (True or False)

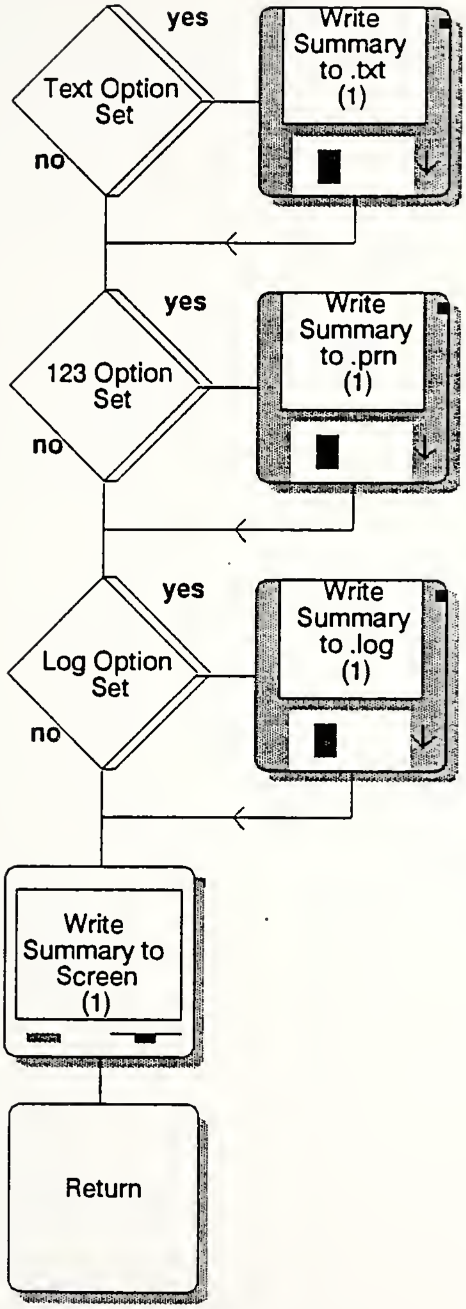
Optional:

- Next 4 repairs in life cycle ("look ahead" option)
- Present worth of repairs

(2) Output file contains only unique information needed by subsequent programs in the system:

- Bridge number and designation
- Repair year and code
- Condition numbers from DTREE
- Exception information from DTREE
- Repair cost, EUAC, and CEF in base year dollars
- Flag to indicate exceptions
- Year of bridge replacement, from life cycle
- Estimated ADT
- New clear deck width after repair

Figure B.6 Write COST Reports



(1) Summary information:

- Number of records successfully processed
- Number of errors
- Total and average cost of projects, EUAC, and CEF

Figure B.7 Write COST Summary

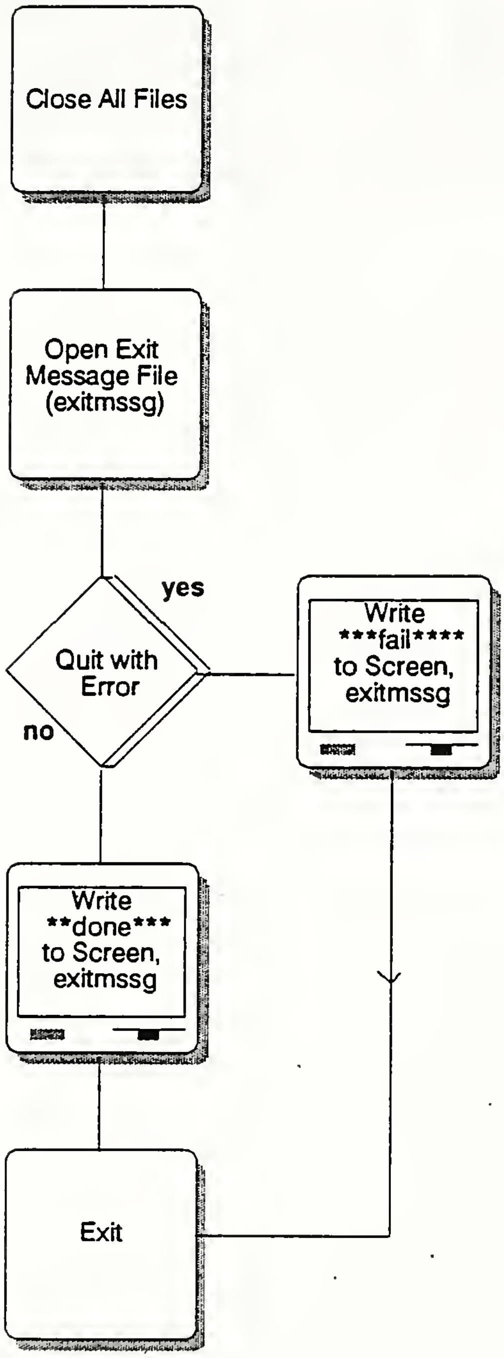


Figure B.8 Quit COST

Appendix C RANK Flow Charts

* Refer to separate flow chart.

(1) One action per bridge has been selected and written to a work file called "Filtered File."

(2) Convert Recording and Coding Guide codes to codes used by the program. For example, assign default values for approach lengths. See Chapter 2 for more information.

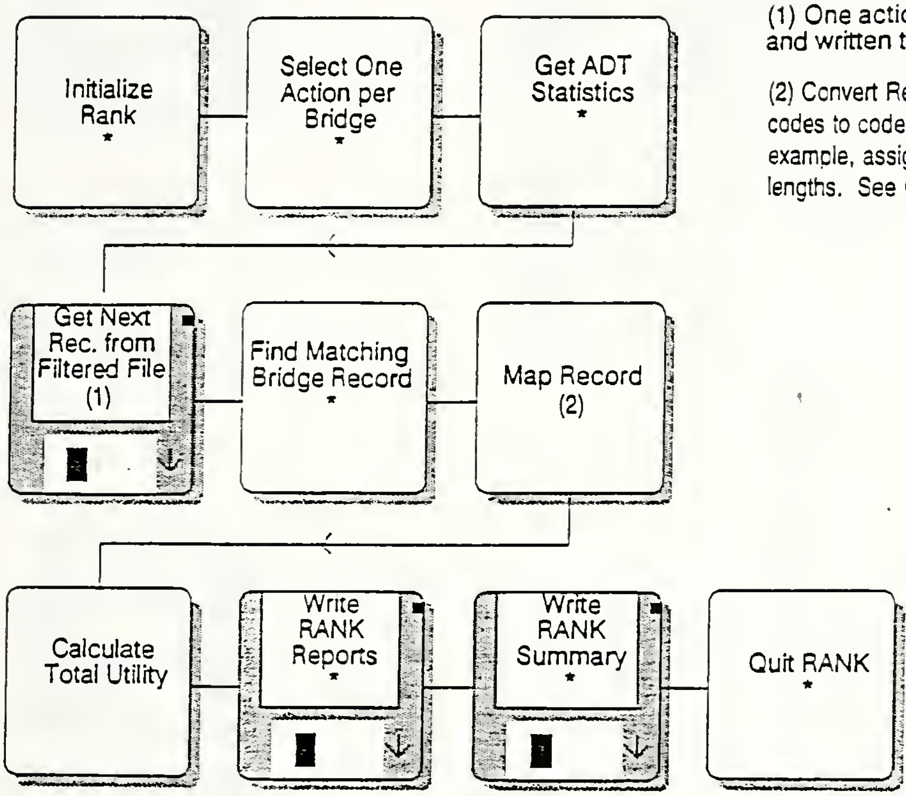
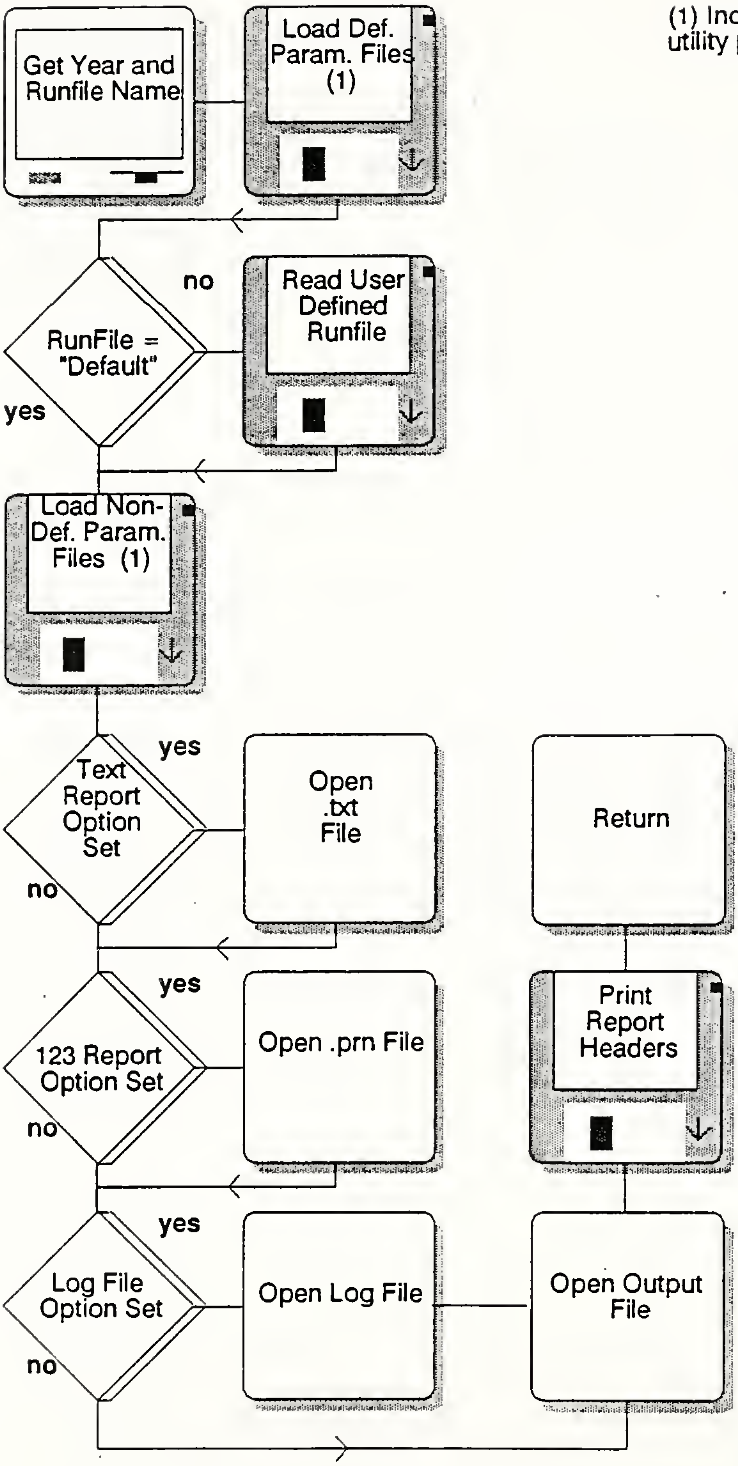
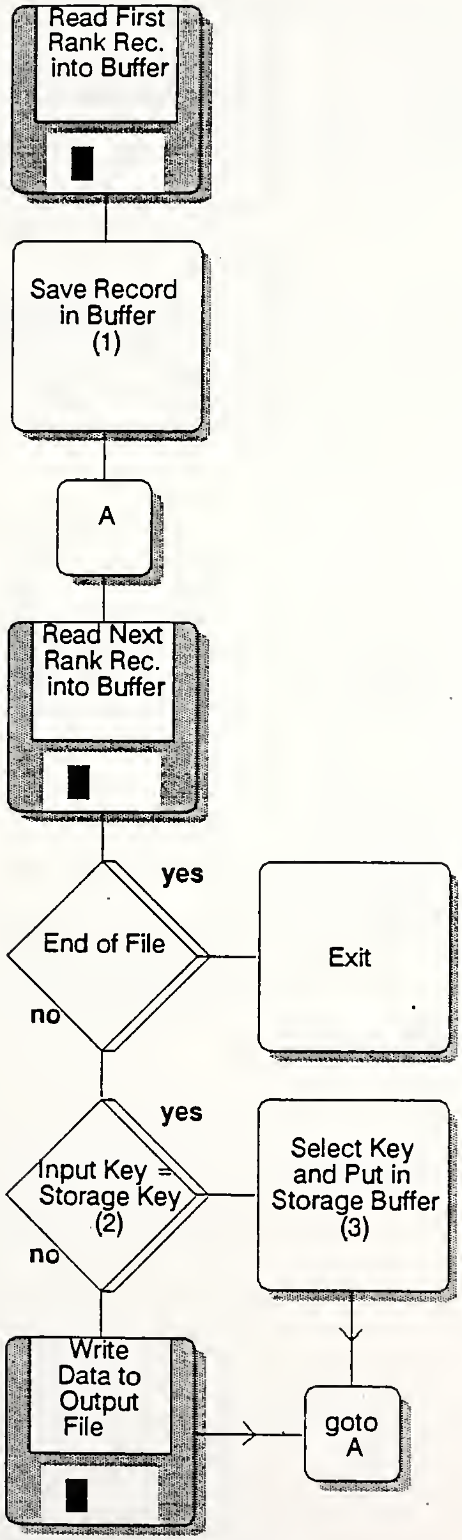


Figure C.1 RANK: Main Program



(1) Includes ranking weight and ranking utility parameter files.

Figure C.2 Initialize RANK

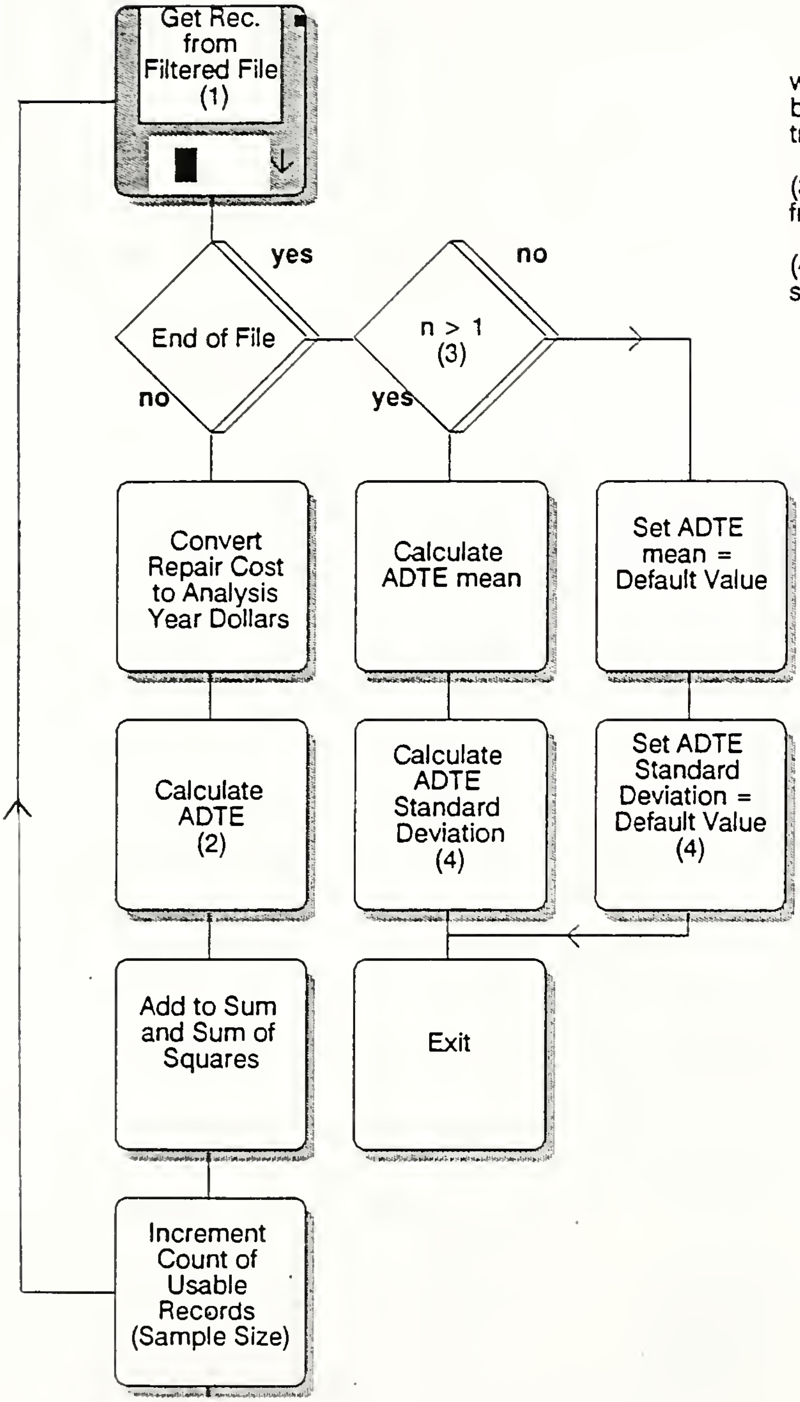


(1) The information read into the rank input common block (input buffer) is copied into an identical block (storage buffer).

(2) The bridge number and designation of the record in the input file are the same as those in the storage buffer.

(3) Select the record with the lowest EUAC.

Figure C.3 Select One Action per Bridge



(1) This file has only one selected action per bridge.

(2) Average Daily Traffic Effectiveness (ADTE) is calculated by:

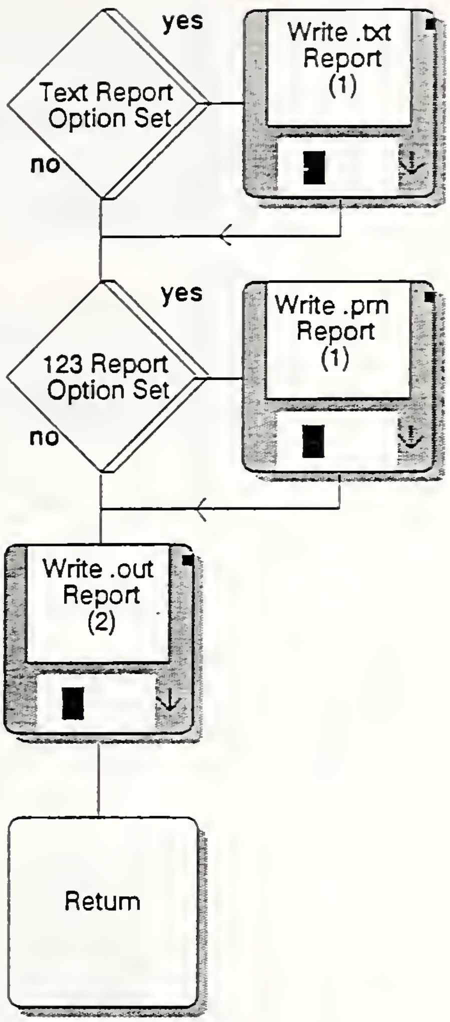
ADTE = 365 * ADT/CEF

where CEF = cost effectiveness factor, in base year dollars and ADT = average daily traffic

(3) n = number of records successfully read from the filtered input file (sample size)

(4) Calculate using sum, sum of squares, and sample size.

Figure C.4 Get ADT Statistics



(1) Reports include:

Standard:
- Bridge number and Designation
- Route number
- District and County
- Repair year
- Repair code and description
- Repair cost, in $1000's, analysis year dollars
- EUAC, $1000's
- Exceptions used (True or False)
- Total disutility before and after repair, in action year
- Change in disutility before and after repair

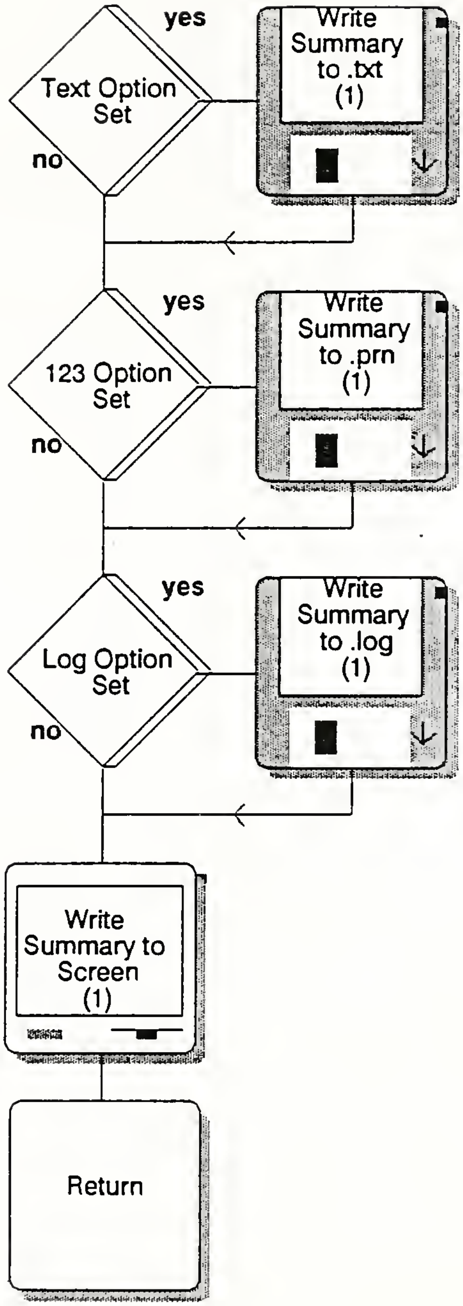
Optional:

- Individual disutilities

(2) Output file contains only unique information needed by subsequent programs in the system:

- Bridge number and designation
- Repair year and code
- Condition numbers, in action year
- Exception information from DTREE
- Repair cost from COST, in analysis year dollars
- Base EUAC and CEF from COST
- Exception flag from COST
- Estimated disutility before and after repair for 2,3,4, and 5 years from analysis year
- Estimated ADT 2,3,4, and 5 years from analysis year

Figure C.5 Write RANK Reports



(1) Summary information:

- Sample size
- Number of actions selected and rejected
- Number of errors
- ADTE mean and standard deviation

Notes

- if fatal errors occur, default values for the mean and standard deviation are used
- the sum of the number of records selected and rejected should equal the sample size

Figure C.6 Write RANK Summary

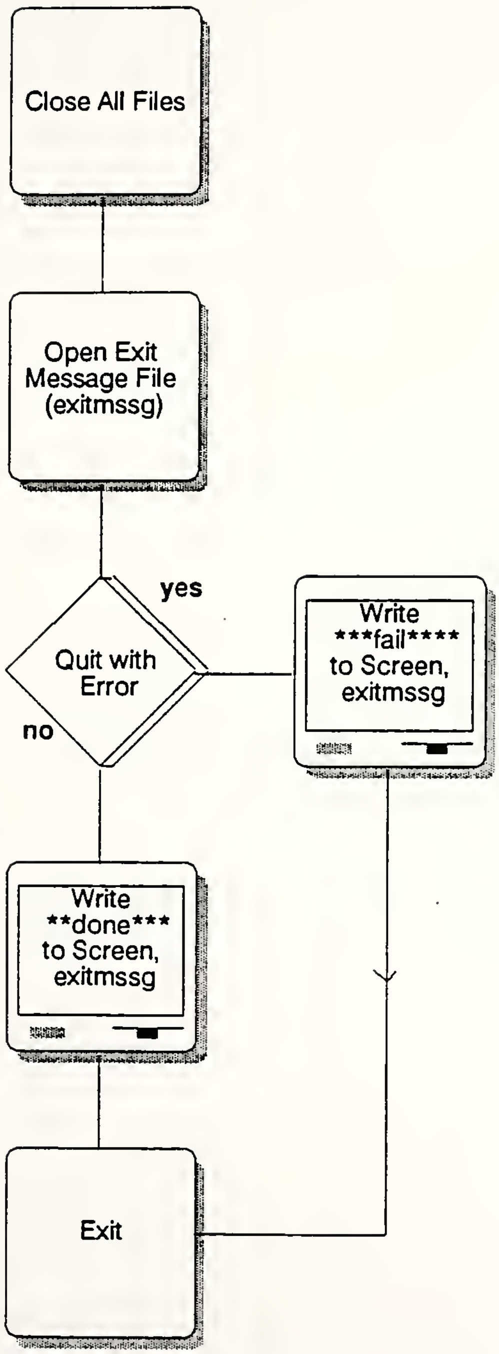
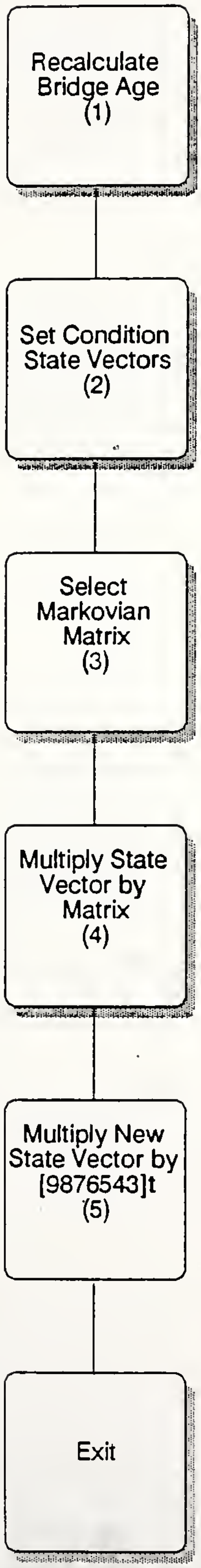


Figure C.7 Quit RANK

Appendix D Flow Charts For Commonly Used Programs



(1) Determine the number of years since the last repair at the time the condition numbers are estimated.

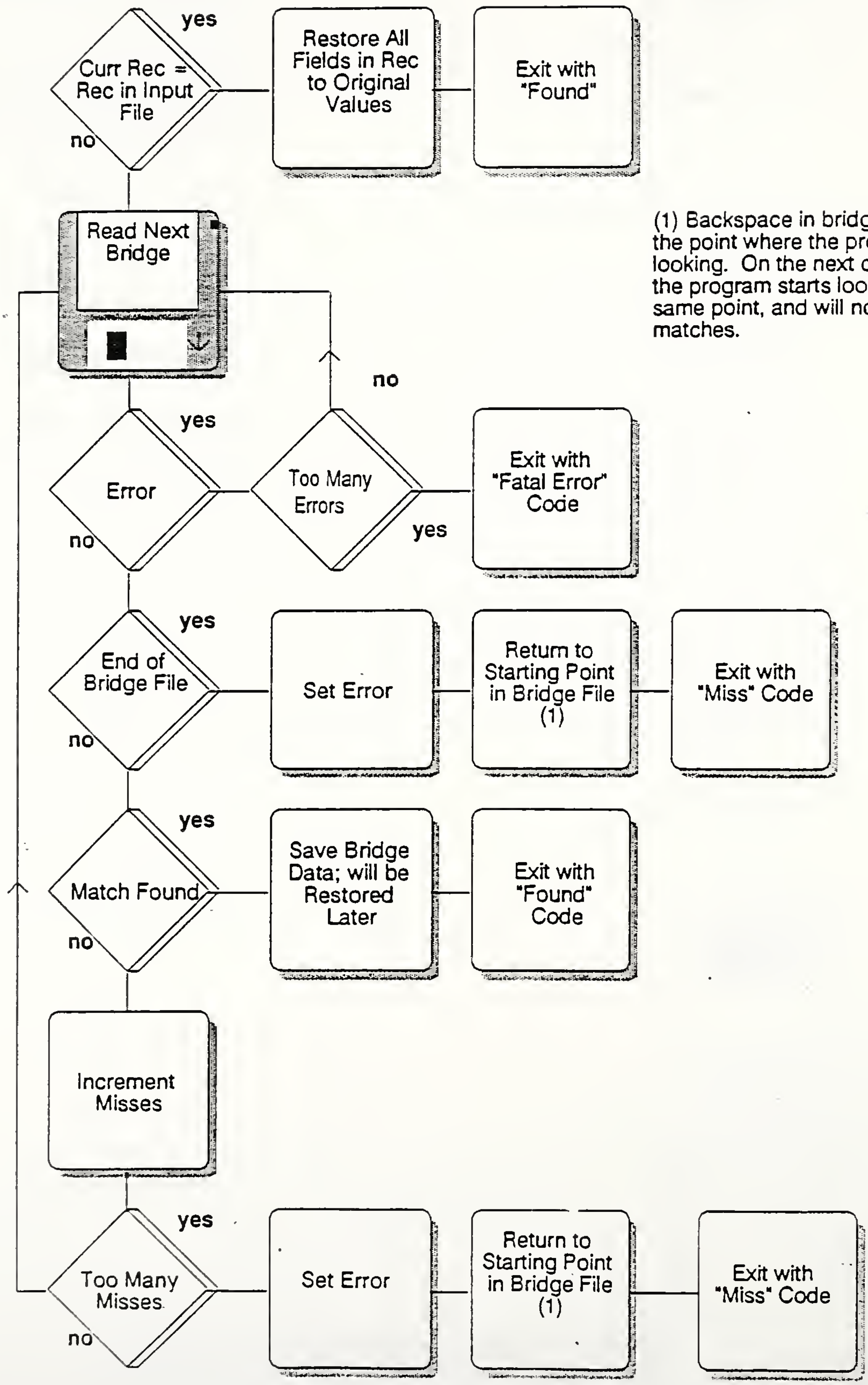
(2) If this is the first projection year (one year after inspection), set state vector to unit vector: [1000000]. However, if the condition number is 5, the initial vector will be [0000100]. Similarly, if the condition number is 7, the initial vector is [0010000]. Refer to FHWA/IN/JHRP-89/13, Volume 6, Section 2.5 for more information.

(3) Select the transition matrix, based on the bridge's age and condition at this point. Refer to the tables in FHWA/IN/JHRP-89/13, Volume 6, Figures 2.6-2.17.

(4) $v' = v * [M]$, where v is the state vector and M is the matrix selected in (3).

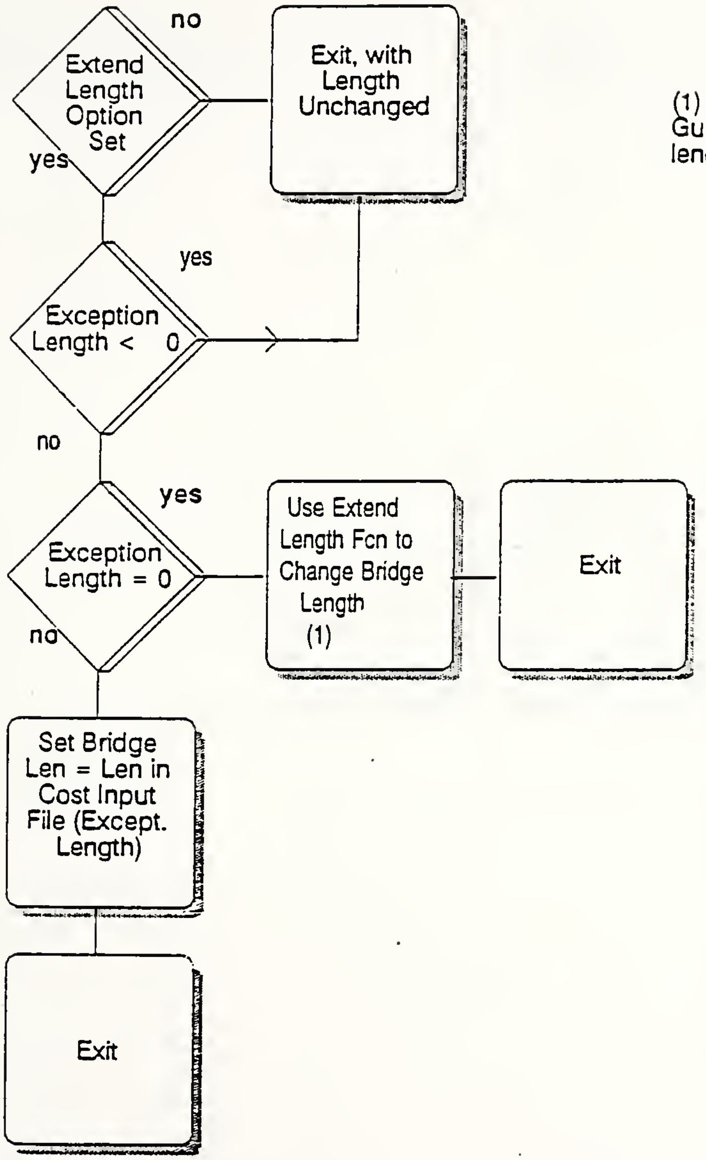
(5) Multiply the current state vector by the transpose of [9876543] and round to the nearest integer. This vector becomes the new condition number.

Figure D.1 Degrade Condition Numbers



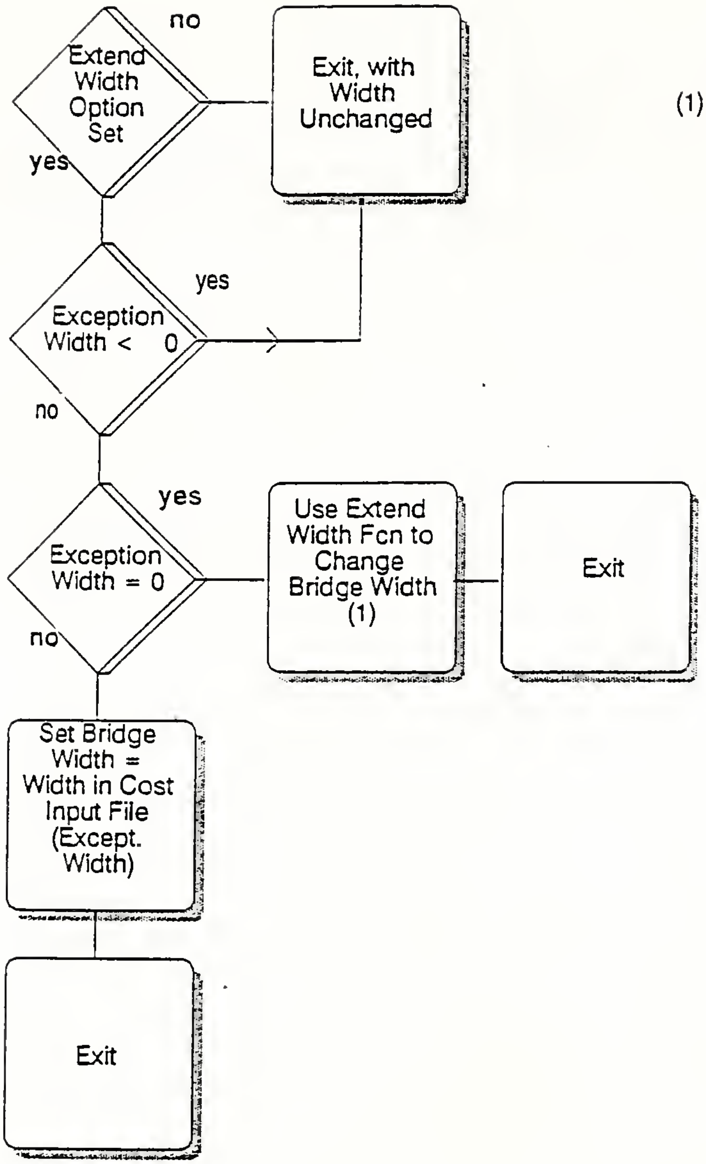
(1) Backspace in bridge database file to the point where the program first started looking. On the next call to this routine, the program starts looking from the same point, and will not overlook any matches.

Figure D.2 Find Matching Bridge Record



(1) See extend.for or Recording and Coding Guide, Item 76, for function used to extend length.

Figure D.3 Extend Bridge Length



(1) See extend.for for more information.

Figure D.4 Extend Bridge Width

Appendix E OPT: Program Operation

The programming style of OPT differs from that of the first three programs (DTREE, COST, and RANK). OPT uses the XMP software developed by Roy A. Marsten to optimize year by year the projects to be selected. For a description of the particular program used, consult "ZOOM Users Manual" 12/92. For a flow chart of the optimization process, including the updating of bridge information, see "The Development of Optimal Strategies for Maintenance, Rehabilitation and Replacement of Highway Bridges, Final Report Vol. 6: Performance Analysis and Optimization," by Yi Jiang and K.C. Sinha, Report No. FHWA/IN/JHRP-89/13, Joint Highway Research Project, School of Civil Engineering, West Lafayette, Indiana. The bridge updating is accomplished with C Language UTILITIES SORT_COST_BY_YEAR.C, MAKE_RANK_OUT.C, and WRITE_OPT_BY2.C.

Appendix F File Formats

F.1 Format for Bridge Database Records

These records are created by extracting data from the main bridge inventory database. Each record is composed of ASCII characters and is terminated by a new line. The bridge records are read by the Fortran function GetNextBridge in the program GETBRIDG.FOR. Refer to Chapter 7 for information on printing the bridge database file.

Table F.1 Bridge Database Format

Record Layout

POS. = the position of the field in the record
ITEM = the Recording and Coding Guide item number
FORMAT = the Fortran format used to read the field

POS.	ITEM	FORMAT	DESCRIPTION
1-4	5d	a4	highway route number
5-7	3	i3	county code

The next 2 fields define the unique bridge key (9 characters)

POS.	ITEM	FORMAT	DESCRIPTION
8-12	---	a5	bridge number
13-16	---	a4	bridge designation
17-18	2	i2	highway dept. district code
19-22	27	i4	year of original construction
23-26	106	i4	last year bridge was repaired
27-28	26	i2	functional class code
29	104	i1	hwy system of inventory route
30-35	29	i6	average daily traffic (ADT)
36-37	28a	i2	number of traffic lanes
38-41	52	f4.1	total deck width (ft)
42-45	51	f4.1	clear deck width (ft)
46-51	49	i6	bridge length (ft)
52-53	54	i2	vertical clearance (ft)
54-55	54	i2	vertical clearance (in)
56	43a	i1	superstructure material
57-58	43b	i2	superstructure construction
59-60	19	i2	detour length (mi)
61	66a	i1	type of load
62-63	66b	i2	inventory rating (tons)
64	58	i1	main deck surface condition rating
65	59	i1	superstructure condition rating
66	60	i1	substructure condition rating
67	68	i1	deck geometry code
68-69	75	i2	proposed work code
70-73	90	i4	date of last inspection (yr/month)

```
1   5    10   15   20   25   30   35   40   45   50   55   60   65   70
----+----+----+----+----+----+----+----+----+----+----+----+----+----+---
000703900652    051928196106200420002038903690000780000102051235445319108
002405600899B   041929199106200380002048704480000300000102061298658  9004
005607801083A   051947198506200130002034302800001620000102081316665  9008
022708901363A   031938197307400150002037403430001262108602021325756359108
003307102037    0419360000         00018001600001321402303007006766358912
I06504902431    031974000011108980008160515750002922204402021295577358912
002405202456AWBL021975199114200340002043704070001562207402011257667  9109
033105003451    041947000007400250002031302800000720000201062365666319002
I06501004216BSBL051958198901101324502042603950001241410102011328775  9005
I06900204540CSBL021961198801101260003047604130002101404402011296667349204
002405204858AWBL021963198602200700002041003800001020000201011216666  9201
I46500605289A   0319711982011      02029202620002121502202042327775  9109
025202406934    051988000007400140002046504350001010000201092368867  9104
001803807558    0219860000062001700020000000000000030000031904236NNNN  9202
002708907710SBL 0319760000         00009900670000841903502008007778  9104
```

Figure F.1 Bridge Database File

F.2 Format of DTREE Output Records

These records contain unique information generated by DTREE that is passed to subsequent programs in the IBMS. Since DTREE is the only program in the system that reads exception files, this information is passed to other programs via an output file with the extension .out. Other information not passed in the output file can be viewed by printing out the bridge database file; refer to Chapter 7 for more information.

These records are written by the subroutine WriteDtreeReports in the file dtree.for and are read by the COST program by the function GetNextCostInput in the file COST.FOR.

Table F.2 DTREE Output File Format

Record Layout

POS. = the position of the field in the record
FORMAT = the FORTRAN format used to read the record

POS.	FORMAT	DESCRIPTION
1-5	a5	bridge number in the bridge database file
6-9	a4	bridge designation in the bridge file
10-13	i4	action year generated by DTREE
14-15	i2	action code generated by DTREE
16	i1	superstructure condition at time of action
17	i1	substructure condition at time of action
18	i1	deck condition at time of action
19-24	i6	exception bridge length from the exception file
25-29	f5.1	exception bridge width from the exception file
30-35	i6	exception approach length
36-37	i2	exception vertical clearance (ft)
38-39	i2	exception vertical clearance (in)

```
1   5    10   15   20   25   30   35   40   45
----+----+----+----+----+----+----+----+----+
00652    199416345     0  0.0     0 0 0
00899B   199410547     0  0.0     0 0 0
01083A   199413555     0  0.0     0 0 0
01083A   199514554     0  0.0     0 0 0
01363A   1994 1755     0  0.0     0 0 0
01363A   1995 2754     0  0.0     0 0 0
01363A   1996 7744     0  0.0     0 0 0
02431    1994 4574     0  0.0     0 0 0
02456AWBL1994 3456     0  0.0     0 0 0
02456AWBL1995 8445     0  0.0     0 0 0
02456AWBL1997 9334     0  0.0     0 0 0
03451    199411443     0  0.0     0 0 0
03451    1995 9343     0  0.0     0 0 0
04216BSBL1994 0567     0  0.0     0 0 0
04540CSBL199415555     0  0.0     0 0 0
04540CSBL199612444     0  0.0     0 0 0
04858AWBL1994 0666     0  0.0     0 0 0
04858AWBL1996 1655     0  0.0     0 0 0
04858AWBL1997 6645     0  0.0     0 0 0
06934    1994 5747     0  0.0     0 0 0
```

Figure F.2 DTREE Output File

F.3 Format for Exception Records

Exception records and their use are described in detail in Chapter 6. Exception records are located by the function GetException in the program getex.for. Records are then read sequentially by the function GetNextException, which is also in GETEX.FOR.

Table F.3 Exception File Format

Record Layout

POS. = the position of the field in the record
FORMAT = the FORTRAN format used to read the record

POS.	FORMAT	DESCRIPTION
1-5	a5	bridge number
6-9	a4	bridge designation
10-11	i2	code for action
12-15	i4	year of action
16-21	i6	new bridge length after next action
22-26	f5.1	new bridge width after next action
27-32	i5	approach length to be used
33-34	i3	vert clear (ft) after next action
35-36	i2	vert clear (in) after next action

```
1   5    10   15   20   25   30   35   40
----+----+----+----+----+----+----+----+
02052CSBL 5   0     0  0.0     0 0 0
02307ASBL-1   0     0  0.0    -1 0 0
02431     -12000     0  0.0     0 0 0
026  A    16   0     0  0.0     0 0 0
04331AWBL13   0     0555.0     0 0 0
04426AWBL13   0     0 -1.0     0 0 0
04461A    16   0   555  0.0     0 0 0
0452A     -1   0     0555.0     0 0 0
04603     -1   0     0 -1.0     0 0 0
04727A    -1   0     0  0.0     0 0 5
04734ANBL-1   0     0  0.0     0 0-1
04751ANBL-1   0     0  0.0   555 0 0
04770A    -1   0     0  0.0     055 0
04780     -1   0     0  0.0     0-1 0
04813B    -1   0     0  0.0     0 0 0
05180BWBL-1  -1    -1 -1.0    -1-1-1
05181BWBL13   0     0  0.0     0 0 0
05271A    16   0    -1  0.0     0 0 0
05484     -1   0   555  0.0     0 0 0
05487ASBL-1   0    -1  0.0     0 0 0
```

Figure F.3 Exception File

F.4 Format for COST Output Records

These records contain unique information generated by COST, as well as information passed from DTREE, that is used in subsequent programs in the system.

These records are written by the subroutine WriteCostReports in the program COST.FOR; they are read by the RANK program using the function GetNextRankInput in the program RANK.FOR.

Table F.4 COST Output File Format

Record Layout

POS. = the position of the field in the record
FORMAT = the FORTRAN format used to read the record

POS.	FORMAT	DESCRIPTION
1-5	a5	bridge number from the bridge database file
6-9	a4	bridge designation
10-13	i4	year of action
14-15	i2	action code
16	i1	superstructure condition at time of action
17	i1	substructure condition at time of action
18	i1	deck condition at time of action
19-24	i6	exception bridge length passed from DTREE via COST
25-29	f5.1	exception bridge width
30-35	i6	exception approach length
36-37	i2	exception vertical clearance (ft)
38-39	i2	exception vertical clearance (in)
40-46	f7.1	cost of project in base year dollars
47-53	f7.1	EUAC in base year dollars, assuming project specified is done in the action year
54-60	f7.3	CEF in base year dollars
61	L1	flag set to TRUE if any exception information was used by the COST program
62	L1	flag set to TRUE if the action specified above was generated from the life cycle model instead of DTREE
63-66	i4	year bridge will be replaced based on the action recommended and the life cycle model
67-72	i6	estimated ADT at the year of action specified
73-77	f5.1	clear deck width after action specified above

```
1   5    10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+
00652     199416345    0  0.0    0 0 0  194.5    19.6 24.187FF1994  4596 36.9
00899B    199410547    0  0.0    0 0 0   49.3     7.7119.076FF2024  4284 44.8
01083A    199413555    0  0.0    0 0 0  174.9    23.6  3.948FF2024  1466 28.0
01083A    199514554    0  0.0    0 0 0  233.2    25.5  3.644FF2025  1510 28.0
01363A    1994 1755    0  0.0    0 0 0   95.7    15.2  7.422FF2024  1641 34.3
01363A    1995 2754    0  0.0    0 0 0  266.0    24.9  4.528FF2025  1691 34.3
01363A    1996 7744    0  0.0    0 0 0  266.0    23.0  4.889FF2026  1743 34.3
02431     1994 4574    0  0.0    0 0 0 2778.1   291.6  2.174FF2039104333157.5
02456AWBL1994 3456     0  0.0    0 0 0  145.3    26.0  6.677FF2039  3720 40.7
02456AWBL1995 8445     0  0.0    0 0 0  173.0    25.9  6.719FF2040  3833 40.7
02456AWBL1997 9334     0  0.0    0 0 0  361.2    32.4  5.358FF2042  4071 40.7
03451     199411443    0  0.0    0 0 0  140.3    15.8 25.528FF2009  2819 28.0
03451     1995 9343    0  0.0    0 0 0  145.6    14.9 26.939FF2010  2905 28.0
04540CSBL199415555     0  0.0    0 0 0  350.0    47.0 10.029FF2039 13379 41.3
04540CSBL199612444     0  0.0    0 0 0  324.1    38.8 12.150FF2041 14206 41.3
04858AWBL1996 1655     0  0.0    0 0 0  128.3    15.0 38.981FF2026  7892 38.0
04858AWBL1997 6645     0  0.0    0 0 0  128.3    13.9 42.084FF2027  8133 38.0
06934     1994 5747    0  0.0    0 0 0   25.0    10.2 10.137FF2024  1532 43.5
```

Figure F.4 COST Output File

F.5 Format for Rank Output Records

These records contain information on project ranking. RANK calculates the disutility of the project before and after repairs are completed. RANK will calculate the before and after disutilities at the recommended year, as well as 2, 3, 4 and 5 years after the year of analysis. The results are stored separately and then passed to OPT. RANK output records are written by the subroutine WriteRankReports, found in the program RANK.FOR.

Table F.5 RANK Output File Format

Record Layout

POS. = the position of the field in the record
FORMAT = the FORTRAN format used to read the record

POS.	FORMAT	DESCRIPTION
1-5	a5	bridge number in the bridge database
6-9	a4	bridge designation
10-13	i4	year of action
14-15	i2	action code
16	i1	superstructure condition at the year of action
17	i1	substructure condition at the year of action
18	i1	deck condition at the year of action
19-24	i6	exception bridge length passed from DTREE via COST
25-29	f5.1	exception clear deck width
30-35	i6	exception approach length
36-37	i2	exception vertical clearance (ft)
38-39	i2	exception vertical clearance (in)
40-46	f7.1	repair cost in analysis year dollars from COST
47-53	f7.1	base EUAC from COST
54-60	f7.3	base CEF from COST
61	L1	exception used flag from COST
62	L1	if TRUE, repair was derived from the life cycle instead of DTREE RANK should not process any records set to TRUE, so all output in this field will be FALSE
63-64	i2	disutility of project if done in year of action
65-68	i4	year of replacement from COST
69-74	i6	estimated ADT in the year of action
75-76	i2	estimated disutility after the project is done (if done in the year of action)
77-78	i2	change in disutility after project completion
79-80	i2	disutility of project 2 years after analysis year
81-82	i2	disutility of project 3 years after analysis year
83-84	i2	disutility of project 4 years after analysis year
85-86	i2	disutility of project 5 years after analysis year
87-88	i2	estimated disutility after project

Table F.5, continued

		completion if done 2 years after year of analysis
89-90	i2	estimated disutility after project completion if done 3 years after year of analysis
91-92	i2	estimated disutility after project completion if done 4 years after year of analysis
93-94	i2	estimated disutility after project completion if done 5 years after year of analysis
95-100	i6	estimated ADT 2 years after year of analysis
101-106	i6	estimated ADT 3 years after year of analysis
107-112	i6	estimated ADT 4 years after year of analysis
113-118	i6	estimated ADT 5 years after year of analysis

```
1   5    10   15   20   25   30   35   40   45   50   55   60   65   70   75   80   85   90   95  100  105  110  115  120
----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+
00652     199416345    0  0.0     0 0 0  206.2    19.6 24.187FF571994  4596 75057575757 7 7 7 7  4596  4735  4880  5028
00899B    199410547    0  0.0     0 0 0   52.3     7.7119.076FF172024  4284 9 817172121 9 91313  4284  4415  4549  4688
01083A    199413555    0  0.0     0 0 0  185.4    23.6  3.948FF262024  146618 82630303018222222  1466  1510  1556  1604
01363A    1994 1755    0  0.0     0 0 0  101.5    15.2  7.422FF212024  164113 82125252513171717  1641  1691  1743  1796
02431     1994 4574    0  0.0     0 0 0 2945.6   291.6  2.174FF22203910433310122222222101010101043331075101107841141 58
02456AWBL1995 8445     0  0.0     0 0 0  183.4    25.9  6.719FF222040  3833 61622222226 6 6 610  3720  3833  3950  4071
03451     1995 9343    0  0.0     0 0 0  154.4    14.9 26.939FF322010  290512203232353812121212  2819  2905  2993  3084
04540CSBL199612444     0  0.0     0 0 0  343.6    38.8 12.150FF362041 1420628 83232363624242828 13379 13787 14206 14639
04858AWBL1997 6645     0  0.0     0 0 0  136.0    13.9 42.084FF232027  813315 815191923 7111115  7433  7659  7892  8133
06934     1994 5747    0  0.0     0 0 0   26.5    10.2 10.137FF222024  153214 82226262614181818  1532  1578  1627  1676
```

Figure F.5 RANK Output File

Appendix G Sample Reports

Mapped INDOT BMS Bridge database file: test.dat Checked

Route	Co.	Br # des	Dis	YrBlt	Fixed	FC	Hwy	ADT	Lns	DkWid	ClWd	Length	Vft	Vin	SM	SC	Det	IR	Ton	DC	SC	sC	DG	WK	IDate	DeckT
0007	39	00652	5	1961	1961	2	2	4200	2	38.9	36.9	78	16	0	2	4	5	1	23	5	4	4	5	31	9108	*
0024	56	00899B	4	1991	1991	2	2	3800	2	48.7	44.8	30	16	0	2	4	6	1	29	8	6	5	8	0	9004	*
0056	78	01083A	5	1985	1985	2	2	1300	2	34.3	28.0	162	16	0	2	4	8	1	31	6	6	6	5	0	9008	*
0227	89	01363A	3	1973	1973	1	4	1500	2	37.4	34.3	126	21	8	2	2	2	1	32	5	7	5	6	35	9108	*
0033	71	02037	4	1936	0	1	0	0	0	18.0	16.0	132	14	2	1	4	0	7	0	6	7	6	6	35	8912	*
I065	49	02431	3	1974	0	1	1	89800	8	160.5	157.5	292	22	4	1	4	2	1	29	5	5	7	7	35	8912	*
0024	52	02456AWBL	2	1991	1991	1	2	3400	2	43.7	40.7	156	22	7	1	4	1	1	25	7	6	6	7	0	9109	*
0331	50	03451	4	1947	0	1	4	2500	2	31.3	28.0	72	16	0	2	1	6	2	36	5	6	6	6	31	9002	*
I069	2	04540CSBL	2	1988	1988	1	1	12600	3	47.6	41.3	210	14	4	1	4	1	1	29	6	6	6	7	34	9204	*
0024	52	04858AWBL	2	1986	1986	1	2	7000	2	41.0	38.0	102	16	0	2	1	1	1	21	6	6	6	6	0	9201	*
I465	6	05289A	3	1982	1982	1	1	0	2	29.2	26.2	212	15	2	2	4	4	2	32	7	7	7	5	0	9109	*
0252	24	06934	5	1988	0	1	4	1400	2	46.5	43.5	101	16	0	2	1	9	2	36	8	8	6	7	0	9104	*
0018	38	07558	2	1986	0	2	2	1700	2	0.0	0.0	30	16	0	1	2	4	2	36	*	*	*	*	0	9202	*
0027	89	07710SBL	3	1976	0	1	0	0	0	9.9	6.7	84	19	3	2	2	0	8	0	7	7	7	8	0	9104	*
I065	10	04216BSBL	5	1989	1989	1	1	13245	2	42.6	39.5	124	14	10	2	4	1	1	32	8	7	7	5	0	9005	*

Figure G.1 Mapped Bridge File

Table G.1 Description of Mapped Bridge Report[1]

Heading	Description
Route	Route number
Co.	County code
Br # des	Bridge number and designation
Dis	District code
YrBlt	Year bridge was constructed
Fixed	Year of last repair
FC	Functional classification
Hwy	Highway system of the inventory route
ADT	Average daily traffic
Lns	Number of lanes
DkWid	Deck width
ClWd	Clear deck width
Length	Length of bridge
Vft	Vertical clearance, ft
Vin	Vertical clearance, in
SM	Superstructure material
SC	Superstructure construction
Det	Detour length
IR	Type of load
Ton	Inventory rating
DC	Deck condition
SC	Superstructure condition
sC	Substructure condition
DG	Deck geometry code
WK	Proposed work code
IDate	Last inspection date
DeckT[2]	Deck type

[1] See Chapter 2, Section 6 for more information.
[2] This field is not used in the current IBMS.

RUNFILE: test.run DECISION TREE REPORT YEAR OF ANALYSIS: 1992

Bridge num des	Route #	Dis	Co. Code	Act Year	Act Code	Action	1992 Cost ($1000)	Work Code	Work Proposed	Ex ?
00652	0007	5	39	1994	16	Replace	206.2	31	Replace Substd	F
00899B	0024	4	56	1994	10	Sub&Sup Rehab	52.3	0		F
01083A	0056	5	78	1994	13	Widen&Deck Rehab	185.5	0		F
01083A	0056	5	78	1995	14	Widen&Deck Repl	247.3	0		F
01363A	0227	3	89	1994	1	Deck Rehab	101.4	35	Rehab Substd	F
01363A	0227	3	89	1995	2	Deck Replace	282.1	35	Rehab Substd	F
01363A	0227	3	89	1996	7	Sub RH Deck Repl	282.1	35	Rehab Substd	F
02431	1065	3	49	1994	4	Sup RH Deck Repl	2945.6	35	Rehab Substd	F
02456AWBL	0024	2	52	1994	3	Super&Deck Rehab	154.1	0		F
02456AWBL	0024	2	52	1995	8	Sub,Sup&Deck RH	183.4	0		F
02456AWBL	0024	2	52	1997	9	Sub RH Sup Repl	383.0	0		F
03451	0331	4	50	1994	11	Sub&Sup RH,Dk RP	148.7	31	Replace Substd	F
03451	0331	4	50	1995	9	Sub RH Sup Repl	154.4	31	Replace Substd	F
04540CSBL	1069	2	2	1994	15	Raise/Lower Pvmt	371.1	34	Widen w/ Repair	F
04540CSBL	1069	2	2	1996	12	Super Replace	343.6	34	Widen w/ Repair	F
04858AWBL	0024	2	52	1994	0	. . . . .	0.0	0		F
04858AWBL	0024	2	52	1996	1	Deck Rehab	136.1	0		F
04858AWBL	0024	2	52	1997	6	Sub&Deck Rehab	136.1	0		F
06934	0252	5	24	1994	5	Substruct Rehab	26.5	0		F

Total Records Successfully Processed 10
Total Errors In Processing Records 0

Decision	Count
Nothing	1
Deck Rehab	2
Deck Replace	1
Super&Deck Rehab	1
Sup RH Deck Repl	1
Substruct Rehab	1
Sub&Deck Rehab	1
Sub RH Deck Repl	1
Sub,Sup&Deck RH	1
Sub RH Sup Repl	2
Sub&Sup Rehab	1
Sub&Sup RH,Dk RP	1
Super Replace	1
Widen&Deck Rehab	1
Widen&Deck Repl	1
Raise/Lower Pvmt	1
Replace	1

Figure G.2 DTREE: Normal Report

RUNFILE: test.run DECISION TREE REPORT YEAR OF ANALYSIS: 1992

Bridge num des	Route #	Dis	Co. Code	Act Year	Act Code	Action	1992 Cost ($1000)	Work Code	Work Proposed	Ex?	Super Cond.	Sub Cond.	Deck Cond.	In Yr	Super @InYr	Sub @InYr	Deck @Inyr
00652	0007	5	39	1994	16	Replace	206.2	31	Replace Substd	F	3	4	5	91	4	4	5
00899B	0024	4	56	1994	10	Sub&Sup Rehab	52.3	0		F	5	4	7	90	6	5	8
01083A	0056	5	78	1994	13	Widen&Deck Rehab	185.5	0		F	5	5	5	90	6	6	6
01083A	0056	5	78	1995	14	Widen&Deck Repl	247.3	0		F	5	5	4	90	6	6	6
01363A	0227	3	89	1994	1	Deck Rehab	101.4	35	Rehab Substd	F	7	5	5	91	7	5	5
01363A	0227	3	89	1995	2	Deck Replace	282.1	35	Rehab Substd	F	7	5	4	91	7	5	5
01363A	0227	3	89	1996	7	Sub RH Deck Repl	282.1	35	Rehab Substd	F	7	4	4	91	7	5	5
02431	I065	3	49	1994	4	Sup RH Deck Repl	2945.6	35	Rehab Substd	F	5	7	4	89	5	7	5
02456AWBL	0024	2	52	1994	3	Super&Deck Rehab	154.1	0		F	4	5	6	91	6	6	7
02456AWBL	0024	2	52	1995	8	Sub,Sup&Deck RH	183.4	0		F	4	4	5	91	6	6	7
02456AWBL	0024	2	52	1997	9	Sub RH Sup Repl	383.0	0		F	3	3	4	91	6	6	7
03451	0331	4	50	1994	11	Sub&Sup RH,Dk RP	148.7	31	Replace Substd	F	4	4	3	90	6	6	5
03451	0331	4	50	1995	9	Sub RH Sup Repl	154.4	31	Replace Substd	F	3	4	3	90	6	6	5
04540CSBL	I069	2	2	1994	15	Raise/Lower Pvmt	371.1	34	Widen W/ Repair	F	5	5	5	92	6	6	6
04540CSBL	I069	2	2	1996	12	Super Replace	343.6	34	Widen W/ Repair	F	4	4	4	92	6	6	6
04858AWBL	0024	2	52	1994	0	. . . .	0.0	0		F	6	6	6	92	6	6	6
04858AWBL	0024	2	52	1996	1	Deck Rehab	136.1	0		F	6	5	5	92	6	6	6
04858AWBL	0024	2	52	1997	6	Sub&Deck Rehab	136.1	0		F	6	4	5	92	6	6	6
06934	0252	5	24	1994	5	Substruct Rehab	26.5	0		F	7	4	7	91	8	6	8

Total Records Successfully Processed 10
Total Errors In Processing Records 0

Decision	Count
Nothing	1
Deck Rehab	2
Deck Replace	1
Super&Deck Rehab	1
Sup RH Deck Repl	1
Substruct Rehab	1
Sub&Deck Rehab	1
Sub RH Deck Repl	1
Sub,Sup&Deck RH	1
Sub RH Sup Repl	2
Sub&Sup Rehab	1
Sub&Sup RH,Dk RP	1
Super Replace	1
Widen&Deck Rehab	1
Widen&Deck Repl	1
Raise/Lower Pvmt	1
Replace	1

Figure G.3 DTREE Condition Report

RUNFILE: test.run DECISION TREE REPORT YEAR OF ANALYSIS: 1992

Bridge num des	Route #	Dis	Co. Code	Act Year	Act Code	Action	1992 Cost ($1000)	Work Code	Work Proposed	Ex?	New Length (ft)	New Width (ft)	New Applen (ft)	Vclear ft	Vclear in	Ex Ac
00652	0007	5	39	1994	16	Replace	206.2	31	Replace Substd	F	0	0.0	0	0	0	0
00899B	0024	4	56	1994	10	Sub&Sup Rehab	52.3	0		F	0	0.0	0	0	0	0
01083A	0056	5	78	1994	13	Widen&Deck Rehab	185.5	0		F	0	0.0	0	0	0	0
01083A	0056	5	78	1995	14	Widen&Deck Repl	247.3	0		F	0	0.0	0	0	0	0
01363A	0227	3	89	1994	1	Deck Rehab	101.4	35	Rehab Substd	F	0	0.0	0	0	0	0
01363A	0227	3	89	1995	2	Deck Replace	282.1	35	Rehab Substd	F	0	0.0	0	0	0	0
01363A	0227	3	89	1996	7	Sub RH Deck Repl	282.1	35	Rehab Substd	F	0	0.0	0	0	0	0
02431	1065	3	49	1994	4	Sup RH Deck Repl	2945.6	35	Rehab Substd	F	0	0.0	0	0	0	0
02456AWBL	0024	2	52	1994	3	Super&Deck Rehab	154.1	0		F	0	0.0	0	0	0	0
02456AWBL	0024	2	52	1995	8	Sub,Sup&Deck RH	183.4	0		F	0	0.0	0	0	0	0
02456AWBL	0024	2	52	1997	9	Sub RH Sup Repl	383.0	0		F	0	0.0	0	0	0	0
03451	0331	4	50	1994	11	Sub&Sup RH,Dk RP	148.7	31	Replace Substd	F	0	0.0	0	0	0	0
03451	0331	4	50	1995	9	Sub RH Sup Repl	154.4	31	Replace Substd	F	0	0.0	0	0	0	0
04540CSBL	1069	2	2	1994	15	Raise/Lower Pvmt	371.1	34	Widen w/ Repair	F	0	0.0	0	0	0	0
04540CSBL	1069	2	2	1996	12	Super Replace	343.6	34	Widen w/ Repair	F	0	0.0	0	0	0	0
04858AWBL	0024	2	52	1994	0	. . . . .	0.0	0		F	0	0.0	0	0	0	0
04858AWBL	0024	2	52	1996	1	Deck Rehab	136.1	0		F	0	0.0	0	0	0	0
04858AWBL	0024	2	52	1997	6	Sub&Deck Rehab	136.1	0		F	0	0.0	0	0	0	0
06934	0252	5	24	1994	5	Substruct Rehab	26.5	0		F	0	0.0	0	0	0	0

Total Records Successfully Processed 10
Total Errors In Processing Records 0

Decision	Count
Nothing	1
Deck Rehab	2
Deck Replace	1
Super&Deck Rehab	1
Sup RH Deck Repl	1
Substruct Rehab	1
Sub&Deck Rehab	1
Sub RH Deck Repl	1
Sub,Sup&Deck RH	1
Sub RH Sup Repl	2
Sub&Sup Rehab	1
Sub&Sup RH,Dk RP	1
Super Replace	1
Widen&Deck Rehab	1
Widen&Deck Repl	1
Raise/Lower Pvmt	1
Replace	1

Figure G.4 DTREE Exception Report

Table G.2 Description of DTREE Report

Heading	Description
Bridge num and des	Bridge number and designation
Route #	Route number
Dis	District code
Co. Code	County code
Act Year	Action year
Act Code	Action code
Action	Action
1992 Cost ($1000)	Base year cost converted to analysis year dollars
Work Code	Work code from NBIS Recording and Coding Guide
Work Proposed	Work proposed by field inspector
Ex? (T or F)	T: Exception record used F: Exception record not used
Super Cond.[1]	Superstructure condition number in analysis year
Sub Cond.[1]	Substructure condition number in analysis year
Deck Cond.[1]	Deck condition number in analysis year
In Yr[1]	Inspection year
Super@InYr, Sub@InYr, and Deck@InYr[1]	Condition numbers in inspection year
New Length (ft)[2]	Replaces length in bridge data base file
New Width (ft)[2]	Replaces width in bridge data base file
New Applen (ft)[2]	Replaces default approach length
Vclear ft and in[2]	Replaces vertical clearance (feet and inches) in bridge data base file
Ex Ac[2]	Replaces any action specified by DTREE

[1] These items appear only if a condition report style is specified.
[2] These items appear only if a exception report style is specified. In addition, if exceptions are not used, as in this example, these columns will be filled with values of zero. See Appendix F.2 for more information.

Table G.2, continued

Action Code Abbreviation[3]	
Rehab, RH	Rehabilitation
Replace, Repl, RP	Replacement
Sup, Super	Superstructure
Sub, Substruct	Substructure
Pvmt	Pavement
Dk	Deck

[3] See Appendix H for more information.

```
RUNFILE: test.run        COST    PROGRAM    REPORT       YEAR OF ANALYSIS: 1992

                      D                                      Yr 1992          CEF   E
 Bridge       Route   i   Co.   Act    Act                    Cost    EUAC    v/$/   x
num    des      #     s   Code  Year   Code      Action      ($1000)  $1000  1000f2  ?
-------------------------------------------------------------------------------------
00652        0007     5    39   1994    16 Replace               0.0    20.8   25.64 F
00899B       0024     4    56   1994    10 Sub&Sup Rehab         0.0     8.2  126.26 F
01083A       0056     5    78   1994    13 Widen&Deck Rehab      0.0    25.0    4.19 F
01083A       0056     5    78   1995    14 Widen&Deck Repl       0.0    27.1    3.86 F
01363A       0227     3    89   1994     1 Deck Rehab            0.0    16.1    7.87 F
01363A       0227     3    89   1995     2 Deck Replace          0.0    26.4    4.80 F
01363A       0227     3    89   1996     7 Sub RH Deck Repl      0.0    24.4    5.18 F
02431        I065     3    49   1994     4 Sup RH Deck Repl      0.0   309.2    2.30 F
02456AWBL    0024     2    52   1994     3 Super&Deck Rehab      0.0    27.6    7.08 F
02456AWBL    0024     2    52   1995     8 Sub,Sup&Deck RH       0.0    27.4    7.12 F
02456AWBL    0024     2    52   1997     9 Sub RH Sup Repl       0.0    34.4    5.68 F
03451        0331     4    50   1994    11 Sub&Sup RH,Dk RP      0.0    16.7   27.07 F
03451        0331     4    50   1995     9 Sub RH Sup Repl       0.0    15.8   28.56 F
04540CSBL    I069     2     2   1994    15 Raise/Lower Pvmt      0.0    49.9   10.63 F
04540CSBL    I069     2     2   1996    12 Super Replace         0.0    41.2   12.88 F
04858AWBL    0024     2    52   2006     3 Super &Deck Rehab     0.0   142.9    7.63 F
04858AWBL    0024     2    52   1996     1 Deck Rehab            0.0    15.9   41.33 F
04858AWBL    0024     2    52   1997     6 Sub&Deck Rehab        0.0    14.8   44.62 F
06934        0252     5    24   1994     5 Substruct Rehab       0.0    10.8   10.75 F
-------------------------------------------------------------------------------------

===== Summary ====
Records Processed:     19
Total Errors     :      0

Total Cost ($1000)      6492.7
Total EUAC ($1000)       720.2
Total CEF  ($1000)       461.6

AVE   Cost ($1000)       341.7
AVE   EUAC ($1000)        37.9
AVE   CEF  ($1000)        24.3
```

Figure G.5 COST Normal Report

RUNFILE: test.run COST PROGRAM REPORT YEAR OF ANALYSIS: 1992

Bridge num	des	Route #	Dis	Co. Code	Act Year	Act Code	Action	Yr 1992 Cost ($1000)	EUAC $1000	CEF v/$/ 1000f2	Ex ?	Year	AC	Yr 1992 Cost ($1000)	Year	AC	Yr 1992 Cost ($1000)	Year	Ac	Yr 1992 Cost ($1000)
00652		0007	5	39	1994	16	Replace	206.2	20.8	25.64	F	2014	3	103.6	2029	3	103.6	2044	16	206.2
00899B		0024	4	56	1994	10	Sub&Sup Rehab	52.3	8.2	126.26	F	2009	3	49.9	2024	16	134.2	2044	3	49.9
01083A		0056	5	78	1994	13	Widen&Deck Rehab	185.5	25.0	4.19	F	2009	3	125.6	2024	16	309.1	2044	3	125.6
01083A		0056	5	78	1995	14	Widen&Deck Repl	247.3	27.1	3.86	F	2010	3	125.6	2025	16	309.1	2045	3	125.6
01363A		0227	3	89	1994	1	Deck Rehab	101.4	16.1	7.87	F	2009	3	106.5	2024	16	297.6	2044	3	106.5
01363A		0227	3	89	1995	2	Deck Replace	282.1	26.4	4.80	F	2010	3	106.5	2025	16	297.6	2045	3	106.5
01363A		0227	3	89	1996	7	Sub RH Deck Repl	282.1	24.4	5.18	F	2011	3	106.5	2026	16	297.6	2046	3	106.5
02431		1065	3	49	1994	4	Sup RH Deck Repl	2945.6	309.2	2.30	F	2009	2	2805.3	2024	3	577.3	2039	16	2019.4
02456AWBL		0024	2	52	1994	3	Super&Deck Rehab	154.1	27.6	7.08	F	2009	2	408.1	2024	3	154.1	2039	16	383.0
02456AWBL		0024	2	52	1995	8	Sub,Sup&Deck RH	183.4	27.4	7.12	F	2010	2	408.1	2025	3	154.1	2040	16	383.0
02456AWBL		0024	2	52	1997	9	Sub RH Sup Repl	383.0	34.4	5.68	F	2012	2	408.1	2027	3	154.1	2042	16	383.0
03451		0331	4	50	1994	11	Sub&Sup RH,Dk RP	148.7	16.7	27.07	F	2009	16	154.4	2029	3	77.0	2044	3	77.0
03451		0331	4	50	1995	9	Sub RH Sup Repl	154.4	15.8	28.56	F	2010	16	154.4	2030	3	77.0	2045	3	77.0
04540CSBL		1069	2	2	1994	15	Raise/Lower Pvmt	371.1	49.9	10.63	F	2009	2	598.3	2024	3	225.9	2039	16	482.5
04540CSBL		1069	2	2	1996	12	Super Replace	343.6	41.2	12.88	F	2011	2	598.3	2026	3	225.9	2041	16	482.5
04858AWBL		0024	2	52	2006	3	Super &Deck Rehab	48.6	7.6	86.65	F	2011	3	142.9	2026	16	250.8	2046	3	142.9
04858AWBL		0024	2	52	1996	1	Deck Rehab	136.1	15.9	41.33	F	2011	3	142.9	2026	16	250.8	2046	3	142.9
04858AWBL		0024	2	52	1997	6	Sub&Deck Rehab	136.1	14.8	44.62	F	2012	3	142.9	2027	16	250.8	2047	3	142.9
06934		0252	5	24	1994	5	Substruct Rehab	26.5	10.8	10.75	F	2009	3	106.1	2024	16	280.8	2044	3	106.1

===== Summary ====
Records Processed: 19
Total Errors : 0

Total Cost ($1000) 6492.7
Total EUAC ($1000) 720.2
Total CEF ($1000) 461.6

AVE Cost ($1000) 341.7
AVE EUAC ($1000) 37.9
AVE CEF ($1000) 24.3

Figure G.6 COST Look Ahead Report

RUNFILE: test.run COST PROGRAM REPORT YEAR OF ANALYSIS: 1992

Bridge num des	Route #	Dis	Co. Code	Act Year	Act Code	Action	PW 1992 Cost ($1000)	EUAC $1000	CEF v/$/ 1000f2	Ex ?
00652	0007	5	39	1994	16	Replace	0.0	20.8	25.64	F
00899B	0024	4	56	1994	10	Sub&Sup Rehab	0.0	8.2	126.26	F
01083A	0056	5	78	1994	13	Widen&Deck Rehab	0.0	25.0	4.19	F
01083A	0056	5	78	1995	14	Widen&Deck Repl	0.0	27.1	3.86	F
01363A	0227	3	89	1994	1	Deck Rehab	0.0	16.1	7.87	F
01363A	0227	3	89	1995	2	Deck Replace	0.0	26.4	4.80	F
01363A	0227	3	89	1996	7	Sub RH Deck Repl	0.0	24.4	5.18	F
02431	I065	3	49	1994	4	Sup RH Deck Repl	0.0	309.2	2.30	F
02456AWBL	0024	2	52	1994	3	Super&Deck Rehab	0.0	27.6	7.08	F
02456AWBL	0024	2	52	1995	8	Sub,Sup&Deck RH	0.0	27.4	7.12	F
02456AWBL	0024	2	52	1997	9	Sub RH Sup Repl	0.0	34.4	5.68	F
03451	0331	4	50	1994	11	Sub&Sup RH,Dk RP	0.0	16.7	27.07	F
03451	0331	4	50	1995	9	Sub RH Sup Repl	0.0	15.8	28.56	F
04540CSBL	I069	2	2	1994	15	Raise/Lower Pvmt	0.0	49.9	10.63	F
04540CSBL	I069	2	2	1996	12	Super Replace	0.0	41.2	12.88	F
04858AWBL	0024	2	52	1996	1	Deck Rehab	0.0	15.9	41.33	F
04858AWBL	0024	2	52	1997	6	Sub&Deck Rehab	0.0	14.8	44.62	F
06934	0252	5	24	1994	5	Substruct Rehab	0.0	10.8	10.75	F

===== Summary ====

Records Processed:	19
Total Errors :	0
Total Cost ($1000)	5252.3
Total EUAC ($1000)	720.2
Total CEF ($1000)	461.6
AVE Cost ($1000)	276.4
AVE EUAC ($1000)	37.9
AVE CEF ($1000)	24.3

Figure G.7 COST Present Worth Report

Table G.3 Description of COST Report

Heading	Description
Bridge num and des	Bridge number and designation
Route #	Route number
Dis	District code
Co. Code	County code
Act Year	Action year
Act Code	Action code
Action	Action
1992 Cost ($1000)	Base year cost converted to analysis year dollars, in thousands
EUAC $1000	Equivalent uniform annual cost, in analysis year dollars
CEF v/$/1000f2	Cost effectiveness factor, in vehicles/$1000/feet^2
Ex? (T or F)	T: Exception record used F: Exception record not used
Year[1]	Year of next repair in life cycle
Ac[1]	Repair code
Yr 1992 Cost ($1000)[1]	Estimated cost, in analysis year dollars
PW 1992 Cost ($1000)[2]	Present worth of repair, in analysis year dollars

[1] These items appear only if a look ahead or present worth, look ahead report style is specified.
[2] These items appear only if a present worth or present worth, look ahead report style is specified.

```
RUNFILE: test.run       RANK    PROGRAM    REPORT       YEAR OF ANALYSIS: 1992

 Bridge    Route   D  Co.   Act   Act                     Cost    EUAC  E Util Util Dlta
num   des    #     s  Code  Year  Code      Action       ($1000) $1000  x Now  Aftr Util
-----------------------------------------------------------------------------------------
00652       0007   5   39   1994   16 Replace             206.2   20.8 F  57     7   50
00899B      0024   4   56   1994   10 Sub&Sup Rehab        52.3    8.2 F  17     9    8
01083A      0056   5   78   1994   13 Widen&Deck Rehab    185.4   25.0 F  26    18    8
01363A      0227   3   89   1994    1 Deck Rehab          101.5   16.1 F  21    13    8
02431       I065   3   49   1994    4 Sup RH Deck Repl   2945.6  309.2 F  22    10   12
02456AWBL   0024   2   52   1995    8 Sub,Sup&Deck RH     183.4   27.5 F  22     6   16
03451       0331   4   50   1995    9 Sub RH Sup Repl     154.4   15.8 F  32    12   20
04540CSBL   I069   2    2   1996   12 Super Replace       343.6   41.1 F  36    28    8
04858AWBL   0024   2   52   1997    6 Sub&Deck Rehab      136.0   14.7 F  23    15    8
06934       0252   5   24   1994    5 Substruct Rehab      26.5   10.8 F  22    14    8
-----------------------------------------------------------------------------------------
SelectOneAction...:     19 actions.    10 selected and     9 rejected.
GetADTStats: 0 errs,  10 ADTEs used to compute mean=27.020     stdev=37.299
```

Figure G.8 RANK Normal Report

```
RUNFILE: test.run                    RANK          PROGRAM        REPORT                  YEAR OF ANALYSIS: 1992

 Bridge    Route   D  Co.   Act   Act                     Cost    EUAC  E Util Util Dlta Cost  Rem.  Strc   Clr. Vert  Inv  Det.
num   des    #     s  Code  Year  Code      Action       ($1000) $1000  x Now  Aftr Util Eff.  Life  Cond   Deck Clr.  Rtng Len.
---------------------------------------------------------------------------------------------------------------------------------
00652       0007   5   39   1994   16 Replace             206.2   20.8 F  57     7   50   51   100   100     9    0    10    0
00899B      0024   4   56   1994   10 Sub&Sup Rehab        52.3    8.2 F  17     9    8    6     0    80     0    0     0    0
01083A      0056   5   78   1994   13 Widen&Deck Rehab    185.4   25.0 F  26    18    8   60     0    60    83    0     0    0
01363A      0227   3   89   1994    1 Deck Rehab          101.5   16.1 F  21    13    8   59     0    60    31    0     0    0
02431       I065   3   49   1994    4 Sup RH Deck Repl   2945.6  309.2 F  22    10   12   61     0    80     0    0     0    0
02456AWBL   0024   2   52   1995    8 Sub,Sup&Deck RH     183.4   27.5 F  22     6   16   59     0    80     0    0     0    0
03451       0331   4   50   1995    9 Sub RH Sup Repl     154.4   15.8 F  32    12   20   49     0   100    83    0     0    0
04540CSBL   I069   2    2   1996   12 Super Replace       343.6   41.1 F  36    28    8   56     0    80    73   83     0    0
04858AWBL   0024   2   52   1997    6 Sub&Deck Rehab      136.0   14.7 F  23    15    8   42     0    80     0    0    20    0
06934       0252   5   24   1994    5 Substruct Rehab      26.5   10.8 F  22    14    8   57     0    80     0    0     0    0
---------------------------------------------------------------------------------------------------------------------------------
SelectOneAction...:     19 actions.    10 selected and     9 rejected.
GetADTStats: 0 errs,  10 ADTEs used to compute mean=27.020     stdev=37.299
```

Figure G.9 RANK Utility Report

Table G.4 Description of RANK Report

Heading	Description
Bridge num and des	Bridge number and designation
Route #	Route number
Ds	District code
Co. Code	County code
Act Year	Action year
Act Code	Action code
Action	Action
Cost ($1000)	Base year cost converted to analysis year dollars, in thousands
EUAC $1000	Equivalent uniform annual cost, in analysis year dollars
Ex? (T or F)	T: Exception record used F: Exception record not used
Util Now	Total disutility of bridge before repair, for action year
Util Aftr	Total disutility after repair, for action year
Dlta Util	Change in total disutility before and after repair, for action year
Cost Eff.[1]	Cost effectiveness disutility function value
Rem. Life[1]	Remaining life disutility function value
Strc Cond[1]	Structural condition disutility function value
Clr. Deck[1]	Clear deck width disutility function value
Vert Clr.[1]	Vertical clearance disutility function value
Inv Rtng[1]	Inventory rating disutility function value
Det. Len.[1]	Detour length disutility function value

[1] These items appear only if a utility report style is specified. In addition, these values represent the disutility before a repair is made.

```
INDOT BMS        DECISION TREE LOG     Runfile:  test.ruYEAR OF ANALYSIS: 1992
-------------------------------------------------------------------------------

---------------------------------------------------------------------------------------

 Total Records Successfully Processed    10
 Total Errors In Processing Records       0
 Decision          Count
 ===============   =====
 Nothing               2
 Deck Rehab            2
 Deck Replace          1
 Super&Deck Rehab      1
 Sup RH Deck Repl      1
 Substruct Rehab       1
 Sub&Deck Rehab        1
 Sub RH Deck Repl      1
 Sub,Sup&Deck RH       1
 Sub RH Sup Repl       2
 Sub&Sup Rehab         1
 Sub&Sup RH,Dk RP      1
 Super Replace         1
 Widen&Deck Rehab      1
 Widen&Deck Repl       1
 Raise/Lower Pvmt      1
 Replace               1
```

Figure G.10 DTREE Log File

```
RUNFILE: test.run       COST    PROGRAM    LOG          YEAR OF ANALYSIS: 1992

-------------------------------------------------------------------------------
 Processing input records...
-------------------------------------------------------------------------------

===== Summary ====
Records Processed:   19
Total Errors     :    0

Total Cost ($1000)     6492.3
Total EUAC ($1000)      720.7
Total CEF  ($1000)      461.6

AVE   Cost ($1000)      341.7
AVE   EUAC ($1000)       37.9
AVE   CEF  ($1000)       24.3
```

Figure G.11 COST Log File

```
RUNFILE: test.run       RANK     PROGRAM     LOG        YEAR OF ANALYSIS: 1992
---------------------------------------------------------------------------------
 SelectOneAction...:    18 actions.    10 selected and     8 rejected.
 GetADTStats: 0 errs,  10 ADTEs used to compute mean=27.020     stdev=37.299
---------------------------------------------------------------------------------
```

Figure G.12 RANK Log File

Appendix H BMS MR & R Activities

December 4, 1991

BMS - MR & R ACTIVITIES

COST ESTIMATES:

The Purdue cost study determined algorithms or cost basis for projects of "Bridge Replacement", "Deck Rehabilitation", and "Deck Replacement". The bridge work category includes other rehabilitation or reconstruction activities than these studied so a cost basis for the other activities must be developed. Without performing a cost study and to proceed with the implementation phase of the Bridge Management System, an expert opinion poll was conducted to determine present default cost estimates for the other activities. A cost study will be performed in the future and the default values will be replaced with determined algorithms. The expert group estimating the cost default values consisted of Jack White, Jim Karr, Danny Wampler, and Robert Woods. The estimated default values for each rehabilitation/reconstruction activity is as follows:

1. Deck Rehabilitation
   Use cost estimate basis as determined by the Purdue study.

2. Deck Replacement
   Use cost estimate basis as determined by the Purdue study.

3. Superstructure Rehab plus Deck Rehabilitation
   Use cost estimate basis as determined by #1 - Deck Rehabilitation listed above and increase five (5) percent.

4. Superstructure Rehab plus Deck Replacement
   Use cost estimate basis as determined by #2 - Deck Replacement listed above and increase five (5) percent.

5. Substructure Rehabilitation
   Use a constant $25,000

6. Substructure Rehab plus Deck Rehabilitation
   Use cost estimate basis as determined by #1 - Deck Rehabilitation listed above.

7. Substructure Rehab plus Deck Replacement
   Use cost estimate basis as determined by #2 - Deck

Replacement listed above.

8. Substructure, Superstructure, and Deck Rehabilitation
   Use cost estimate basis as determined by #1 - Deck Rehabilitation listed above and increase twenty five (25) percent.

9. Substructure Rehabilitation and Superstructure Replacement
   Use cost estimate basis as determined by the Purdue study for a bridge replacement project.

10. Substructure Rehabilitation plus Superstructure Rehabilitation
    Use cost estimate basis as determined by #1 - Deck Rehabilitation listed above and increase ten (10) percent.

11. Substructure Rehab, Superstructure Rehab plus Deck Replacement
    Use cost estimate basis as determined by #4 - Superstructure Rehabilitation plus Deck Replacement listed above and increase five (5) percent.

12. Superstructure Replacement
    Use cost estimate basis as determined by the Purdue study for a bridge replacement project minus the substructure cost and minus 60 percent of the approach cost.

13. Bridge Widening plus Deck Rehabilitation
    Use 60 percent of the cost estimate basis as determined by the Purdue study for a bridge replacement project.

14. Bridge Widening plus Deck Replacement
    Use 80 percent of the cost estimate basis as determined by the Purdue study for a bridge replacement project.

15. Raise Bridge/Lower Pavement
    Use a constant $350,000

16. Bridge Replacement
    Use cost estimate basis as determined by the Purdue study.

17. Culvert Replacement
    Use a constant $200,000

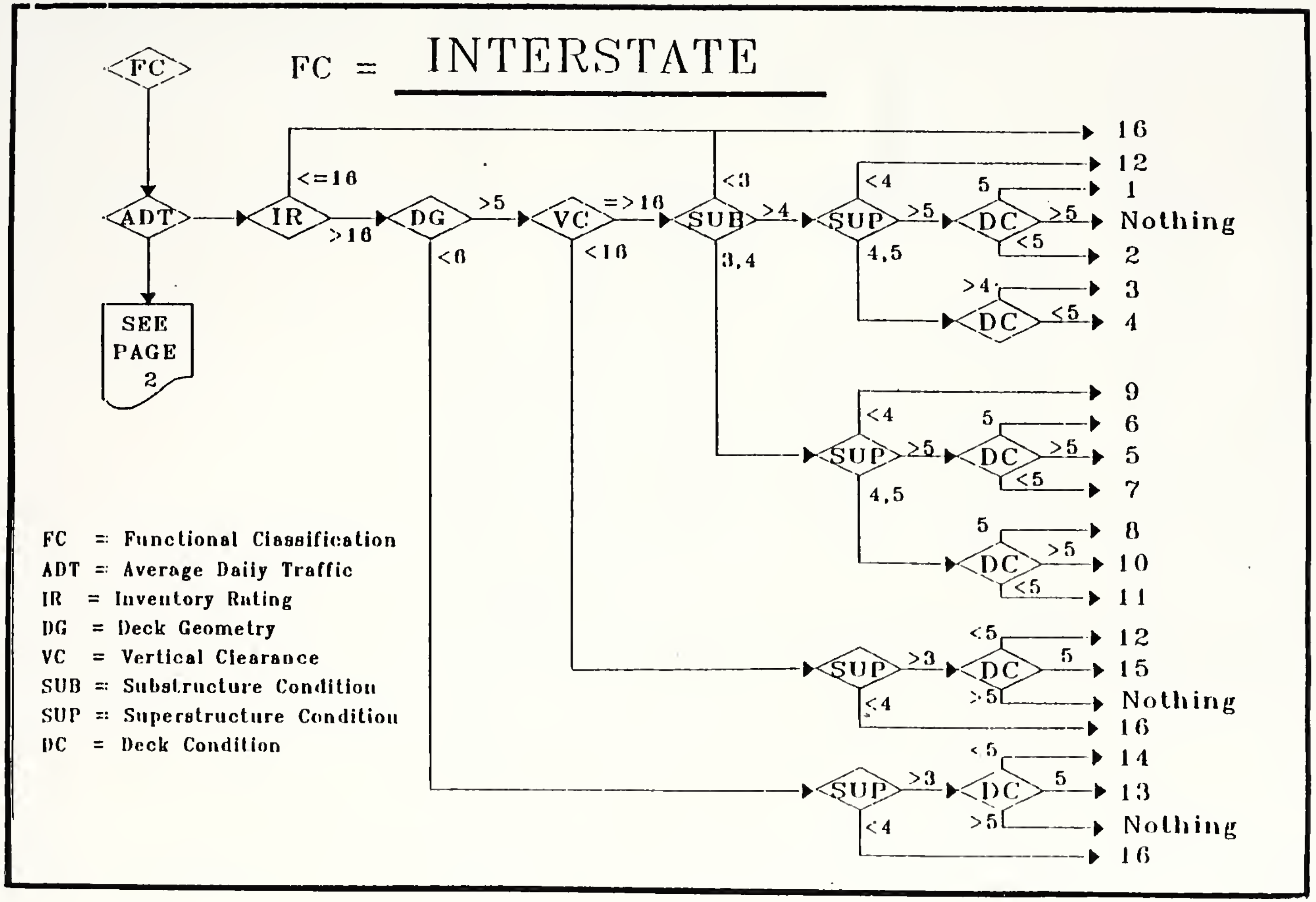


Figure H.1 BMS - Condition Rating Default Decision Tree

Table H.1 Action Code Definitions

Action Code	Action
1	Deck Rehabilitation
2	Deck Replacement
3	Superstr Rehab + Deck Rehab
4	Superstr Rehab + Deck Replacement
5	Substr Rehab
6	Substr Rehab + Deck Rehab
7	Substr Rehab + Deck Replacement
8	Substr, Superstr and Deck Rehab
9	Substr Rehab + Superstr Replacement
10	Substr Rehab + Superstr Rehab
11	Substr Rehab, Superstr Rehab + Deck Replacement
12	Superstructure Replacement
13	Bridge Widening + Deck Rehabilitation
14	Bridge Widening + Deck Replacement
15	Raise Bridge/Lower Pavement
16	Bridge Replacement
17	Culvert Replacement

APPENDIX I COMPILING THE PROGRAMS

I.1 COMPILING FORTRAN PROGRAMS

If modifications are made to any of the programs, the programs must be re-compiled. It is recommended that all four programs be re-compiled, since some of the routines are used by more than one program; for example, DEGRADE.FOR is used by both.

A file called MAKEFILE has been set up to simplify the compiling process; see Figure I.1. This file establishes the necessary program and library links and parameters used by the programs. The makefile is used in conjunction with a program called NMAKE.EXE, which is a utility program in Microsoft C, version 6.0.

To compile and the user must be in the directory \bridge\prog\source\fortran. At the prompt, type "nmake" and the program name. For example, to compile dtree, type "nmake dtree." After compilation is finished, the new executable file (i.e., DTREE.EXE) must be copied to the directory \bridge\prog. Another alternative would be to compile the three programs simultaneously. A batch file called COMPILE.CMD has been created, which compiles all three programs and then copies the executable files to the appropriate directory; simply type "compile" at the prompt.

I.2 Compiling C Programs

The majority of the programs found in the IBMS Tools are written in Microsoft C. When modifications are made to these programs, the program Programmer's Workbench (pwb.exe) can be

used, which is a utility program in Microsoft C.

To compile a program, the user must be in the directory \bridge\prog\source\c. Type "pwb" at the prompt and open the appropriate program. First, make sure that the library is set for OS/2, by pressing Alt-O for the Options menu, and then C for C Compiler Options; under C Libraries, highlight OS/2 with the mouse.

After making modifications, compile the program by pressing Alt-M for the Make menu and then R for Rebuild All. Exit pwb.exe by pressing Alt-F4 and then copy the new program (.exe) to the directory \bridge\prog.

```
# Makefile for FORTRAN programs on the PC

.SUFFIXES: .for
ff=fortran

# Rule to make a FORTRAN program

.for.obj:
        $(ff) $*.for $(FFLAGS)

# Fortran flags
# -------------
FFLAGS = /n /y /L > out

# Macros to link fortran programs with
FLIB=c:\fortran\plib\fortrae.lib
FLIBDOS=c:\fortran\rlib\fortrae.lib

L=/PMTYPE:NOVIO
M=/PMTYPE:VIO

# Head of path for common blocks
# ------------------------------
c = \bridge\common

# MACROS
# ------

cost1 = bdata.obj
cost2 = cost.obj costest.obj econ.obj error.obj extend.obj
cost3 = getbridg.obj map.obj readcost.obj readdoll.obj readlife.obj
cost4 = misc.obj
cost5 = skip.obj readrun.obj

costo1 = $c\run.cmn $c\bridge.cmn $c\costinp.cmn $c\map.cmn $c\unit.cmn
costo2 = $c\dollar.cmn $c\option.cmn $c\unit.cmn $c\life.cmn
costo3 = $c\linfo.cmn $c\costout.cmn $c\text.cmn $c\option.cmn

dtree1 = dtree.obj error.obj getbridg.obj getdecis.obj getex.obj
dtree2 = readrun.obj readtree.obj findend.obj readdoll.obj
dtree3 = econ.obj degrade.obj
dtree4 = misc.obj bdata.obj
dtree5 = map.obj skip.obj costest.obj

dtreeo1 = $c\except.cmn $c\dtreeout.cmn $c\dtreeout.cmn $c\option.cmn
dtreeo2 = $c\run.cmn $c\bridge.cmn $c\text.cmn $c\unit.cmn $c\dollar.cmn

rank1 = rank.obj error.obj readrun.obj readwt.obj readutil.obj    misc.obj
rank2 = econ.obj utility.obj skip.obj getbridg.obj readdoll.obj map.obj
rank3 = bdata.obj degrade.obj extend.obj
rank4 = compute.obj delta.obj

ranko1 = $c\unit.cmn $c\option.cmn $c\run.cmn $c\dollar.cmn $c\rankbuf.cmn
ranko2 = $c\bridge.cmn $c\rankout.cmn $c\rankinp.cmn $c\text.cmn

prbrid = map.obj error.obj getbridg.obj prbridge.obj bdata.obj misc.obj

prexe  = printexe.obj error.obj getex.obj bdata.obj misc.obj

zoom5 = zoom1.obj zomaps.obj zdrive.obj zolp.obj hpsort.obj zoheur.obj
zoom2 = zobab.obj zinc.obj zmisc.obj zfeas.obj ztour.obj zpandc.obj mpsw.obj
zoom3 = zospec.obj xlp1.obj xlp2.obj xlp3.obj xdualsm.obj xdata.obj xrang.obj
zoom4 = ildata.obj xpara.obj la05.obj xlp4.obj

opt1 = optmain.obj makecmn.obj deter.obj wres.obj rrdata.obj inxmp.obj
opt2 = xml2.obj hpsort.obj lopen.obj

put100 = put100.obj
```

Figure I.1 Makefile

```
# Main Programs
# -------------

chop: chop.obj misc.obj
	link $L chop.obj misc.obj , chop.exe, map.out, $(FLIB), nul.def;cost:  $(cost1) $(cost2)
$(cost3) $(cost4) $(cost5)
	link $L /SE:200 $(cost1) $(cost2) $(cost3) $(cost4) $(cost5), cost.exe, map.out, $(FLIB),
	nul.def;
dtree: $(dtree1) $(dtree2) $(dtree3) $(dtree4) $(dtree5)
	link $L /SE:200 $(dtree1) $(dtree2) $(dtree3) $(dtree4) $(dtree5), dtree.exe, map.out,
	$(FLIB), nul.def;

prbridge: $(prbrid)
	link $L $(prbrid), prbridge.exe, map.out, $(FLIB), nul.def;

printexe: $(prexe)
	link $L $(prexe) , printexe.exe, map.out, $(FLIB), nul.def;

rank:  $(rank1) $(rank2) $(rank3) $(rank4)
	link $L /SE:200 $(rank1) $(rank2) $(rank3) $(rank4),rank.exe, map.out, $(FLIB), nul.def;

zoom: $(zoom5) $(zoom2) $(zoom3) $(zoom4)
     link $L /SE:300 $(zoom5) $(zoom2) $(zoom3) $(zoom4), zoom.exe, map.out, $(FLIB), nul.def;

opt: $(opt1) $(opt2)
     link $L /SE:200 $(opt1) $(opt2), opt.exe, map.out, $(FLIB), nul.def;

put100: $(put100)
     link $L $(put100), put100.exe, map.out, $(FLIB), nul.def;

# Program Modules
# ---------------

# bcost.obj: $c\cost.cmn

bdata.obj: $c\dollar.cmn $c\dtree.cmn $c\life.cmn $c\map.cmn \
	   $c\run.cmn $c\utility.cmn $c\weight.cmn

# bdollar.obj: $c\dollar.cmn

# bdtree.obj: $c\dtree.cmn

# blife.obj: $c\life.cmn

# bmap.obj: $c\map.cmn

# brun.obj: $c\run.cmn

# butility.obj: $c\utility.cmn

# bweight.obj: $c\weight.cmn

compute.obj: $c\bridge.cmn $c\rankinp.cmn $c\weight.cmn

cost.obj: $(costo1) $(costo2) $(costo3)

costest.obj: $c\cost.cmn $c\map.cmn $c\bridge.cmn $c\life.cmn

degrade.obj: $c\pmatrix.cmn $c\bridge.cmn $c\map.cmn

delta.obj: $c\delta.cmn

dtree.obj: $(dtreeo1) $(dtreeo2)

econ.obj: $c\dollar.cmn $c\life.cmn $c\bridge.cmn $c\map.cmn

error.obj: $c\global.cmn $c\option.cmn $c\unit.cmn
```

Figure I.1, continued

```
extend.obj: $c\bridge.cmn $c\costinp.cmn $c\map.cmn

# findend.obj: nothing

getbridg.obj: $c\bridge.cmn $c\unit.cmn

getdecis.obj: $c\bridge.cmn $c\dtree.cmn

getex.obj: $c\except.cmn

map.obj: $c\bridge.cmn $c\map.cmn

printexe.obj: $c\except.cmn $c\unit.cmn
        $(ff) printexe.for $(FFLAGS) /W

prbridge.obj: $c\bridge.cmn

rank.obj: $(ranko1) $(ranko2)

readcost.obj: $c\cost.cmn

readdoll.obj: $c\dollar.cmn

readlife.obj: $c\life.cmn

readrun.obj: $c\run.cmn

readtree.obj: $c\dtree.cmn

readwt.obj: $c\weight.cmn

readutil.obj: $c\utility.cmn

#skip.obj: nothing

utility.obj: $c\utility.cmn

makecmn.obj: $c\filedir.cmn

optmain.obj: $c\filedir.cmn

wres.obj: $c\filedir.cmn

rrdata.obj: $c\filedir.cmn
```

Figure I.1, continued

Appendix J Subroutine and Function Definitions

Table J.1 Subroutine Definitions

Subroutine	Program (.for)	Description
ComputeRanks	compute	calculates total disutility before and after repair, as well as 2,3,4, and 5 years from analysis year
GetUtilities	compute	calculates disutilities for each disutility function
NormalizeWeights	compute	normalizes weights so that they sum to 1.0
Cycle	cost	computes costs for one life cycle
EuacEtc	cost	calculates equivalent uniform annual cost (EUAC) and cost effectiveness factor (CEF)
FullCycle	cost	uses life cycle model to determine what repairs should be done and when
MakeCostReports	cost	formats data for output files
PrintCostHeaders	cost	prints headers for reports
QuitCost	cost	closes files and exits
WriteCostReports	cost	writes reports to files (.out, .txt, .log, .prn)
WriteCostSummary	cost	prints summary of program execution
CopyMatrixRowToP	degrade	copies selected row from GetPMatrix to P
Degrade	degrade	uses Markovian process to estimate condition numbers after inspection year

Table J.1, continued

Subroutine	Program (.for)	Description
GetPMatrix	degrade	selects transition matrix, based on age, material, and function class
MultQbyP	degrade	multiplies vector Q by P
OverlayEx1on2	dtree	if information in primary exception record (except.cmn) is 0, information from secondary exception record (except2.cmn) is used
PrintDtreeHeaders	dtree	prints headers for reports
QuitDtree	dtree	closes files and exits
RestoreException	dtree	restores exception records from buffer
SaveException	dtree	stores exception records in buffer
SelectDecisions	dtree	saves repair code only when it occurs for first time
WriteDtreeReports	dtree	writes reports to files (.out, .txt, .log, .prn)
WriteDtreeSummary	dtree	prints summary of program execution
LogError	error	prints errors found in /global/ block
LogMessage	error	writes errors to log file (.log)
SetError	error	sets global error message and code for main programs

Table J.1, continued

Subroutine	Program (.for)	Description
LengthenBridge	extend	extends bridge length by exception record or EstNewLength function
WidenBridge	extend	extends bridge width by exception record or EstNewWidth function
ListErrors	getbridg	writes errors to log file (.log)
MapRecord	map	replaces fields in bridge record with values recognized by program
PutLine	misc	determines length of a line for reports
StripPathHead	misc	removes full path name for report printing
PrintBridgeHeader	prbridge	prints headers for report
PrintExceptionHeader	printexe	prints headers for report
WriteExceptionReport	printexe	prints report for exception records
CopyRankInputToBuffer	rank	copies data from rankinp.cmn to rankbuf.cmn
MakeRankReports	rank	formats data for output files
PrintRankHeaders	rank	prints headers for reports
QuitRank	rank	closes files and exits
WriteRankReports	rank	writes reports to files (.out, .txt, .log, .prn)

Table J.1, continued

Subroutine	Program (.for)	Description
WriteRankSummary	rank	prints summary of program execution
Tack	readrun	splices character strings together to eliminate blanks
Skip	skip	skips n number of lines when reading a file

Table J.2 Function Definitions

Function	Program (.for)	Description
GetRank	compute	computes bridge's total disutility based on condition numbers, bridge properties, and recommended repair
CostInit	cost	opens and reads files needed for program execution
GetNextCostInput	cost	reads next bridge record
GetPW	cost	computes present worth of repairs and maintenance costs
Rehab	costest	calculates rehabilitation costs, based on deck area and patching percentage
RepairCost	costest	calculates cost, in $1000's, for recommended repair
Replace	costest	calculates replacement cost, assuming 100' approach length
YearlyMaintCostEst	costest	estimates yearly maintenance cost
QTimesRTranspose	degrade	multiplies vector Q by transpose of [9 8 7 6 5 4 3]
DeltaDeckCondition	delta	represents the change in deck condition after a repair

Table J.2, continued

Function	Program (.for)	Description
DeltaServiceLife	delta	represents the change in service life after a repair
DeltaSubCondition	delta	represents the change in substructure condition after a repair
DeltaSuperCondition	delta	represents the change in superstructure condition after a repair
DtreeInit	dtree	opens and reads files needed for program execution
BaseYrToYr	econ	converts dollars from base year to specified year (typically the analysis year)
MaintCost	econ	determines the maintenance costs for the life cycle model
PerpetualWorth	econ	computes the life cycle cost in perpetuity
PresentWorth	econ	calculates the present worth of costs
PresentWorthMaint	econ	determines the present worth of maintenance costs
PWGradient	econ	calculates a gradient cost for maintenance

Table J.2, continued

Function	Program (.for)	Description
UniformCost	econ	calculates uniform annual cost for maintenance
YearToYear	econ	converts dollars from one year to another
YrToBaseYr	econ	converts dollars from specified year to base year
EstNewLength	extend	estimates new bridge length after a replacement
EstNewWidth	extend	estimates new clear deck width
FutureADT	extend	estimates future average daily traffic
FindEnd	findend	finds end of file
CheckRec	getbridg	checks bridge records for errors
FindMatch	getbridg	finds matching bridge record
GetNextBridge	getbridg	reads next bridge record
PrintBridge	getbridg	prints bridge records
GetDecision	getdecis	uses decision tree to recommend a repair
CheckExceptionFile	getex	checks to see if file is sorted and entries are in correct format

Table J.2, continued

Function	Program (.for)	Description
GetException	getex	loads exception records, if available
GetExceptionIndex	getex	finds location of bridge number record that corresponds to bridge number key passed to the routine
GetNextException	getex	reads next exception record
IsRehab	map	determines if repair code is a rehabilitation
IsReplace	map	determines if repair code is a replace
IsWiden	map	determines if repair code affects bridge width
MapApplen	map	uses default approach length if none specified
MapFunctClass	map	converts functional class of bridge to one used by program
MapSST	map	determines superstructure type; currently not used
MapSuperConst	map	converts superstructure type code to one used by program
MapSuperMaterial	map	converts superstructure material code to one used by program

Table J.2, continued

Function	Program (.for)	Description
MapVClear	map	uses default vertical clearance if none specified
MapWorkProposed	map	converts "work proposed" field to a code recognized by program
MapYearBuilt	map	stores most recent repair year
GetADTStats	rank	computes mean and standard deviation for cost effectiveness factor
GetNextRankInput	rank	reads next record from rank input file to rankinp.cmn buffer
RankInit	rank	opens and reads files needed for program execution
SelectOneActionPerBridge	rank	if more than one repair is recommended, repair with lowest EUAC is selected
WriteRankInputBuffer	rank	writes information from rankbuf.cmn block to unit u
ReadCostParam	readcost	reads parameters for estimating repair costs
ReadDollarTable	readdoll	reads dollar conversion data
ReadLifeCycle	readlife	reads parameters for life cycle cost model

Table J.2, continued

Function	Program (.for)	Description
ReadRunFile	readrun	loads input and output file names and options
ReadTree	readtree	loads parameters for decision tree
ReadUtilityFunctions	readutil	reads parameters used for disutility functions
ReadWeights	readwt	loads weights used for project ranking
ClearDeckWidthFunction	utility	computes disutility based on clear deck width
DetourLengthFunction	utility	computes disutility based on detour length or cost of temporary run-around
EffectiveWFactor	utility	computes economic effectiveness factor
OperatingRatingFunction	utility	computes disutility based on operating rating (in tons)
OverallRank	utility	computes total disutility for bridge
RemainingLifeFactor	utility	computes disutility based on remaining service life
StructuralConditionFactor	utility	computes disutility based on minimum condition number from deck, substructure, and superstructure condition
VerticalClearanceFunction	utility	computes disutility based on vertical clearance

COVER DESIGN BY ALDO GIORGINI